www.farelettronica.com
n. 339 – settembre 2013

Voltmetro a diodi LED

Come scegliere i potenziometri digitali

Timer multifunzione

RASPBERRY CYCLETTE

Rendi smart la tua cyclette con Raspberry Pi

Intervallometro fotografico

Levitatore magnetico

Sommario

339 - settembre 2013

ABBIAMO BISOGNO DEL TUO AIUTO!
Facci conoscere le tue preferenze votando il tuo articolo preferito! **CLICCA QUI**

È online il nuovo sito di Fare Elettronica! Visitalo subito!

Editoriale

Pedalare!

Diete, integratori, pozioni miracolose… ci si inventa di tutto pur di dimagrire, ma niente è meglio di un buon esercizio fisico. Esistono in commercio una moltitudine di attrezzi tecnologicamente avanzati, ma dai costi veramente proibitivi. Da qui la nostra idea di dotare di intelligenza una comunissima ed economica cyclette da casa. Il cuore dell'applicazione è Raspberry PI e la sua applicazione sulla cyclette ha dato dei risultati veramente interessanti! Scopriteli nell'articolo di pagina 13 e magari inviateci i vostri miglioramenti visto che l'unico limite in questo caso è solo la fantasia. In questo numero troverete anche un interessante progetto che sfida le leggi della fisica: un levitatore magnetico! Come è nostra abitudine vi invitiamo ad utilizzare questo e gli altri progetti della rivista, come base per la realizzazione di applicazioni sempre più complesse. Siamo sicuri delle vostre abilità manuali, per cui vi invitiamo sempre a segnalare le vostre realizzazioni sulla nostra pagina Facebook: www.facebook.com/farelettronica.

Buona lettura!

Maurizio Del Corso

DIRETTORE RESPONSABILE
Antonio Cirella

DIRETTORE TECNICO
Maurizio Del Corso

Segreteria di redazione
Giorgia Generali

Comitato scientifico
Simone Masoni (Microtest), Francesco Picchi (Microtest), Massimo Rovini (Università degli Studi di Pisa).

Art director
Patrizia Villa

Hanno collaborato a questo numero
Angelo de Bartolo
Mario Mottula
Marco Mangiat
Antonio Giannico
Grazia Ancona
Giovanni di Maria

Direzione e redazione
INWARE Edizioni srl
Via Giotto, 7 - 20032 Cormano (MI)
Tel. 02.66504794
Fax 02.42101817
info@inwaredizioni.it
www.inwaredizioni.it
Redazione: fe@inwaredizioni.it

Pubblicità per l'Italia
Agostino Simone
Tel. +39 347 2230684
media@inwaredizioni.it

Europe and Americas
Elisabetta Rossi
Tel. +39 328 3245956
international@inwaredizioni.it

Asia
Cybermedia Communications Inc.
Tel. +886-(0)2-2691-2785
asia@inwaredizioni.it

Rest of the world
Inware Edizioni srl
+39 02 66504794
info@inwaredizioni.it

Stampa
Prontostampa
Via Redipuglia 150
24045 Fara Gera d'Adda (BG)

Distribuzione
Parrini & C s.p.a.
Via di Santa Cornelia, 9
00060 Formello (RM)

Ufficio abbonamenti
INWARE Edizioni srl
Via Giotto, 7 - 20032 Cormano (MI)
Per informazioni, sottoscrizione o rinnovo dell'abbonamento:
abbonamenti@inwaredizioni.it
Tel. 02.66504794 - Fax. 02.42101817
L'ufficio abbonamenti è disponibile telefonicamente dal lunedì al venerdì dalle 14,30 alle 17,30.
Tel. 02.66504794
Fax 02.42101817

• Membership MAKER:
11 numeri di Fare Elettronica in pdf € 29.99
• Membership MAKER FULL:
11 numeri di Fare Elettronica in pdf + Bonus Pack Maker € 39.99

Autorizzazione alla pubblicazione
Tribunale di Milano n.647
del 17/11/2003

Collaborare con FARE ELETTRONICA
Le richieste di collaborazione vanno indirizzate all'attenzione di Maurizio Del Corso (m.delcorso@inwaredizioni.it) e accompagnate, se possibile, da una breve descrizione delle vostre competenze tecniche e/o editoriali, oltre che da un elenco degli argomenti e/o progetti che desiderate proporre.

Elenco inserzionisti

MikroElektronika
Visegradska, 1A - 11000 Belgrade
Tel. +381 11 3628830 - www.mikroe.com

Teledyne LeCroy
via E. Mattei Valecenter 1/C - 30020 Marcon (VE)
Tel. 041/5997011 - www.lecroy.com

Grifo
Via dell'Artigiano 8/6 -
40016 San Giorgio Di Piano (BO)
Tel. 051-892052 - www.grifo.it

Mesago Messemanagement
RotebuehlstraÃŸe 83 - 85 - 70178 Stuttgart,
Tel. +49 711 61946-19 - www.mesago.com/sps

Millennium Dataware
Corso Repubblica 48 - 15057 Tortona (AL)
Tel. 0131 860254 - www.mdsrl.it

Blu Nautilus
Piazzale Cesare Battisti 22/E - 47921 Rimini (RN)
Tel. 0541-439575 - www.blunautilus.it

Fiera Millenaria
Via Fiera Millenaria, 13 - 46023 Gonzaga (MO)
Tel. 0376/58.098 - www.fieramillenaria.it

Elettroshop
Via Giotto, 7 - 20032 Cormano (MI)
Tel. 02 66504755 - www.elettroshop.com

Tecnoimprese
Via Console Flaminio, 19 - 20134 (MI)
Tel. 02 2101111 – www.fortronic.it

News

Dispositivo per il prolungamento della durata della batteria DC/DC buck boost

Linear Technology presenta l'LTC3330, una soluzione completa di energy harvesting di regolazione che fornisce fino a 50mA di corrente di uscita continua per prolungare la durata della batteria quando è disponibile energia raccolta. L'LTC3330 non richiede corrente di alimentazione dalla batteria se viene fornita potenza regolata al carico dall'energia raccolta e soli 750nA se alimentato dalla batteria in assenza di carico. L'LTC3330 integra un alimentatore di energy harvesting ad alta tensione e un convertitore DC/DC buck-boost sincrono alimentato da una batteria a cella principale per creare una singola uscita non interrompibile per le applicazioni di energy harvesting come quelle delle reti di sensori wireless. L'alimentatore di energy harvesting, costituito da un raddrizzatore a ponte a onda intera che accetta ingressi AC o DC e da un convertitore buck ad alta efficienza, raccoglie energia da fonti piezoelettriche (AC), solari (DC) o magnetiche (AC). L'ingresso a cella principale alimenta un convertitore buck-boost che funziona da 1,8V a 5,5V all'ingresso quando non è disponibile energia raccolta per regolare l'uscita a prescindere dal fatto che l'ingresso sia superiore, inferiore o uguale all'uscita. L'LTC3330 passa automaticamente alla batteria quando la fonte di energy harvesting non è più disponibile.

www.linear.com/product/LTC3330

AVX introduce un nuovo condensatore al tantalio elettrolitico umido

AVX Corporation ha introdotto il nuovo condensatore elettrolitico al tantalio umido con terminale assiale serie TWA-E. Presentandosi in un assemblaggio con camicia al tantalio e basetta che assicura una tenuta ermetica in grado di soddisfare i requisiti di shock e vibrazioni previsti dalle norme CECC 30 202-01, la serie mostra un alto livello di CV in quattro dimensioni di capsula (T1-T4). Specificato per 25-125 V e 15-1500 µF, i condensatori serie TWA-E sono idonei a essere utilizzati nelle applicazioni quali: filtratura, alimentato-

News

ri di potenza DC-DC, convertitori di potenza, immagazzinamento dell'energia di massa, potenza d'impulso, radar, trasmettitori RF, comandi laser Prodotti in Europa secondo le norme CECC 30 202-001 (EN 130202), i condensatori serie TWA-E detengono il livello di ispezione E per il testing secondo le procedere CECC. Specificata per l'impiego in una gamma delle temperature da -55 °C a 125 °C, la serie è disponibile con o senza tubetto isolante e può essere fornita con terminazioni in SnPb 60/40 o prive di piombo, in puro stagno.

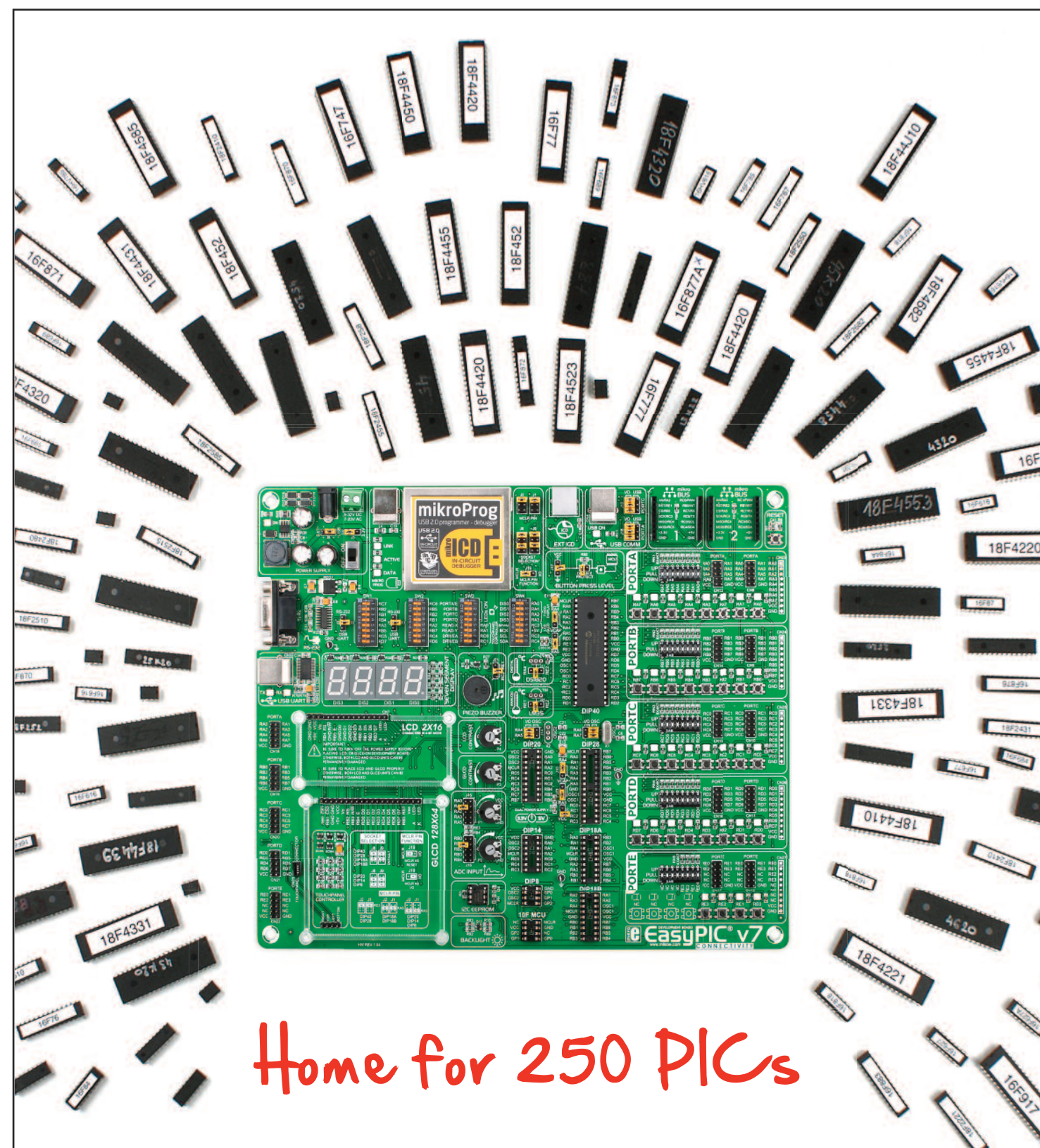
www.avx.com

SL - Data GSM: Serial Data Logger con Modem GSM

SL-Data GSM è un prodotto in grado di raccogliere diversi tipi di segnali, provenienti dal campo, di salvarli su una scheda di memoria del tipo SD o MMC e di trasferirli, su richiesta, ad un sistema esterno tramite una comunicazione seriale senza fili, su rete GSM. Questo prodotto sfrutta l'esperienza, acquisita negli anni dalla Grifo, nello sviluppo dei sistemi di Data Logger. L'apparecchiatura acquisisce e salva diverse grandezze analogiche e digitali in modo semplice, sicuro ed economico. Consente, inoltre, di poter, successivamente, esaminare e/o elaborare le informazioni salvate anche in un altro luogo, senza dover fisicamente raggiungere la sede dell'installazione.

Periferiche da Campo
Controllore e Linee di I/O
Modem GSM
Rete GSM
PC
Modem GSM
Cellulare
Rete GSM

Tutti i dati raccolti sono memorizzati sulle schede di memoria gestite dalla SDI 02, ovvero le SD e le MMC, con un formato FAT16 e rimangono sempre a disposizione.
I dati raccolti possono essere correlati temporalmente tramite un orologio interno (RTC) e possono essere trasferiti ad un PC, o qualsiasi unità esterna, tramite una comunicazione seriale asincrona sia del tipo cablato che senza fili. I dati raccolti sono disponibili sotto forma di file in formato ASCII, organizzati in modo da essere aperti con Excel.
L'SL - Data GSM è distribuito da Grifo - Via dell'Artigiano 8/6 - 40016 San Giorgio di Piano (BO)

www.grifo.it/

News

La famiglia Fairchild Semiconductor di driver low-side ad alta velocità

Le applicazioni destinate al mercato automobilistico che utilizzano MOSFET di potenza necessitano di gate driver con picchi elevati di corrente e bassa impedenza di uscita. Le serie di gate driver low-side ad alta velocità FAN31xx_F085 e FAN32xx_F085 di Fairchild Semiconductor qualificate per il settore automotive offrono flessibilità per alimentatori e altre applicazioni di commutazione MOSFET ad alta efficienza proponendo una vasta selezione di combinazioni di prestazioni e funzionalità per creare design compatti, affidabili e altamente efficienti.
Questi driver qualificati AEC-Q100 sono disponibili in package S08 nelle varianti con soglie di ingresso TTL o CMOS. Ciò assicura la migliore compatibilità possibile con qualsiasi design. I driver hanno due ingressi per ciascun canale (Dual-input con logica + e –, Inverting and Enable, o Non-inverting and Enable) garantiscono flessibilità progettuale.

www.fairchildsemi.com

Mouser fornisce i LED mid-power LUXEON 3535 2D Philips Lumileds

Mouser Electronics, Inc. ha annunciato la disponibilità a magazzino dei LED mid power LUXEON 3535 2D Philips Lumileds. Questo nuovo package di chip a due LED con un'efficienza fino a 140 lm/W può ridurre del 50% il numero di LED nelle lampade retrofit.
Philips Lumileds ha arricchito la famiglia di LED LUXEON Mid-Power 3535 con i nuovi LUXEON 3535 2D Mid-Power. LUXEON 3535 2D combina due chip LED in un package per consentire di ridurre il numero di LED e il costo di progettazione delle lampade retrofit, i faretti per uso generico e le lampade industriali a luce diffusa. I progettisti dei sistemi di illuminazione possono sfruttare il vantaggio di un package con un alto numero di lumen ad elevato rendimento, utilizzando un numero di chip nettamente inferiore per apparecchio. I nuovi LED mid-power vengono offerti con un ampio spettro di temperature di colore (2700K-6500K) e un CRI minimo di 80. La linea LUXEON 3535 2D funziona a 0,5 – 1W, fornendo fino a 130 lumen in un formato 3535 compatto. Con una potenza nominale di 100 mA, LUXEON 3535 2D può arrivare a 200 mA max. Questo LED supporta i requisiti di certificazione ENERGY STAR fino a 25.000 ore a 85° C e 150 mA.

www.mouser.com/new/Philips-Lumileds/philipslumileds-Luxeon3535/

Power Integrations collabora con Qualcomm nello sviluppo della tecnologia per la carica rapida di dispositivi mobili

Power Integrations ha introdotto il CHY100, il primo circuito integrato per caricabatteria CA-CC che permette ai progettisti di dispositivi mobili di implementare il protocollo QuickCharge 2.0 sviluppato da Qualcomm. Lanciato agli inizi di quest'anno, QuickCharge 2.0 consente agli utenti di caricare dispositivi mobili a velocità sino al 75% superiore rispetto all'uso della tecnologia convenzionale.

Impiegato in combinazione con i circuiti integrati con switch CA-CC di Power Integrations, il CHY100 incorpora tutti gli elementi necessari per aggiungere la funzionalità Quick Charge 2.0 ai caricabatteria CA-CC. Il CHY100 rileva i comandi inviati da un dispositivo compatibile con Quick-Charge 2.0, come un microtelefono cellulare, e regola l'uscita del caricabatteria CA-CC per consentire l'erogazione di una potenza maggiore alla batteria del dispositivo. Quando è collegato a un dispositivo alimentato senza la funzionalità Quick Charge 2.0, il CHY100 disabilita automaticamente la funzionalità di tensione/potenza più elevate per garantire il funzionamento in sicurezza. Vari test hanno confermato che smartphone dotati di batterie ad alta capacità possono essere caricati completamente in un tempo brevissimo, sino a un'ora, utilizzando QuickCharge 2.0 – un miglioramento notevole rispetto ai cicli di carica di quattro ore caratteristici dei convenzionali caricabatteria a tensione costante.

www.powerint.com/CHY100

News

Toshiba presenta un fotorelè ultracompatto a bassa corrente di attivazione

Toshiba Electronics Europe lancia un nuovo fotorelè a semiconduttore (fotoaccoppiatore con uscita a MOSFET) che combina una bassissima corrente di accensione con un formato ultra-compatto. Il nuovo TLP175A è particolarmente adatto in sistemi che richiedono commutazioni altamente affidabili e una maggiore efficienza, con risparmio di spazio su scheda.

Il nuovo fotorelè di Toshiba rappresenta un'alternativa più piccola, più affidabile, più durevole ed energeticamente meno onerosa rispetto ai relè meccanici. I possibili impieghi comprendono sistemi di sicurezza, macchine per l'automazione industriale, strumenti di misura come contatori elettronici e accessori. Le distanze superficiali e in aria di soli 5,0 mm e lo spessore di isolamento interno di 0,4 mm (min) contribuiscono a una tensione di isolamento minima pari a 3750 Veff.
Il microcontenitore SO6 a 4 contatti del fotorelè comprende un LED a infrarossi otticamente accoppiato a un fotoMOSFET. Le dimensioni del contenitore sono pari a soli 7,0 x 3,7 x 2,1 mm. 1 mA è la corrente richiesta per attivare il LED e accendere il fotorelè.
Con una configurazione normalmente aperta (1-Form-A), il TLP175A offre una tensione di uscita nello stato di OFF pari a 60 V e una corrente massima nello stato di ON pari a 100 mA.

www.toshiba-components.com

I nuovi oscilloscopi digitali ISO-TECH di RS Components assicurano misurazioni accurate a costi contenuti

RS Components ha introdotto la nuova serie IDS-6000A-U di oscilloscopi digitali a due canali per usi generici firmata ISO-TECH. Con una larghezza di banda che spazia da 70 a 150 MHz, la serie IDS6000A-U è dotata di un display LCD TFT di 5,7 pollici studiato per migliorare e semplificare le misurazioni. Con un design ergonomico e compatto, e un peso ridotto a soli 2,5 kg, questa serie permette un campionamento real-time di 1 GS/s e fino a 25 GS/s su segnali continui con registrazione fino a 4.000 punti, per assicurare la massima flessibilità agli utenti. Molteplici modalità di acquisizione e fino a 27 funzioni di misura automatiche, consentono all'utente di misurare accuratamente le proprietà delle diverse forme d'onda.
La serie IDS-6000A-U consente il controllo da remoto e di acquisire dati attraverso la porta USB o da altre interfacce. La serie supporta PictBridge che permette la stampa diretta, via USB, tramite una stampante compatibile. La tecnologia MemoryPrime permette di ottenere 2 Mpoint di dati sulla forma d'onda, ottimizzando il rapporto tra lunghezza di memoria e velocità di campionamento. I vantaggi di questa tecnologia sono disponibili utilizzando le funzioni Horizontal Page Skip e Set Time.
Grazie ad un menu ad albero per visualizzare in maniera semplice le varie funzioni, la serie IDS-6000A-U soddisfa i requisiti di laboratori, scuole, controllo qualità, test di prodotto, riparazioni, sviluppo prodotti e debugging.

www.rs-components.com

Eventi

ZERO EMISSION ROME 2013

Giunto alla sua sesta edizione, ZEROEMISSION ROME è l'evento di riferimento per tutte le aziende e gli operatori interessati allo sviluppo delle energie rinnovabili, all'emission trading e alla sostenibilità ambientale in Italia e nel grande e promettente mercato del bacino del Mediterraneo. ZEROEMISSION ROME 2013 è l'insieme di eventi specializzati dedicati all'energia eolica, all'energia fotovoltaica, al solare termodinamico, all'emission trading, cambiamenti climatici e CCS, agroenergie e biocarburanti. Insieme occuperanno ben quattro grandi padiglioni di Fiera di Roma su un'area di oltre 40.000 metri quadri.

Dove: Roma • Quando: 9-11 Ottobre 2013 • Orari: dalle 9.30 alle 18.30 • Organizzazione: ZeroEmission • info: www.zeroemissionrome.eu

ROME MAKER FAIRE

La Maker Faire è il più Grande Show al mondo dedicato all'innovazione, creatività e alle idee, un evento per famiglie e amanti del "Do it Yourself" ma anche un evento per celebrare il Movimento dei Maker. La Maker Faire di Roma – The European Edition è la prima Maker Faire Europea realizzata, organizzata e promossa da Asset-Camera con il supporto del Tecnopolo – due realtà del sistema camerale romano- in collaborazione con Arduino e la rivista Make. Tra gli obiettivi degli organizzatori, oltre a quello di far scoprire al grande pubblico il movimento dei Maker, c'è quello di rendere Roma il centro di questa comunità a livello Europeo

Dove: Roma • Quando: 5-6 ottobre 2013 • Orari: dalle 9.00 alle 18.00 • Organizzazione: Asset Camera • info: www.makerfairerome.eu

AUTOMAZIONE IN FIERA

SAVE è un appuntamento innovativo che unisce una parte espositiva in fiera ad una forte componente formativa. Area espositiva dove incontrare agli stand i principali leader di settore, centinaia di convegni e workshop accessibili per gli operatori qualificati, cinque eventi internazionali e un evento speciale in contemporanea, gli operatori professionali accedono gratuitamente alla mostra e usufruiscono di tutti i servizi. L'esigenza percepita oggi è rendere adeguate le modalità fieristiche, renderle attuali e sempre più proficue per creare una fiera sull'automazione e sulla strumentazione utile agli operatori.

Dove: Verona • Quando: 29-30 ottobre 2013 • Orari: dalle 9.00 alle 18.00 • Organizzazione: EIOM • info: www.exposave.com

Sei MAKER, SMART o GENIUS?

Entra anche tu nel mondo dell'elettronica con Inware Edizioni!

CLUB MEMBERSHIP

Club Light

accesso e/o download agli articoli (centinaia) del Club di Firmware

Club Full

accesso e/o download agli articoli (centinaia) del Club di Fare Elettronica

MAKER MEMBERSHIP

- 11 nuovi numeri di Fare Elettronica in PDF
  (i numeri andranno in coda all'eventuale abbonamento esistente)

Bonus Pack Maker (solo per la card Maker Full)

- accesso e/o download agli articoli (centinaia) del Club di Fare Elettronica
- un ebook a scelta* (Smartcard, CPLD, Display LCD, PIC, Operazionali, alimentatori, linguaggio C, Basic per PIC, Elettronica Analogica)
- una raccolta delle annate di Fare Elettronica in PDF a scelta (dal 2003 al 2012)

SMART MEMBERSHIP

- 11 nuovi numeri della rivista digitale Firmware
  (i numeri andranno in coda all'eventuale abbonamento esistente)
- tutto l'archivio delle riviste Firmware pubblicate mensilmente dal febbraio 2010

Bonus Pack Smart (solo per la card Smart Full)

- accesso e/o download agli articoli (centinaia) del Club di Firmware
- un ebook a scelta* (Smartcard, CPLD, Display LCD, PIC, Operazionali, alimentatori, linguaggio C, Basic per PIC, Elettronica Analogica)
- una raccolta delle annate di Firmware in PDF (dal 2006 al 2012)

GENIUS MEMBERSHIP

- 11 nuovi numeri di Fare Elettronica in PDF
- 11 nuovi numeri della rivista digitale Firmware
  (i numeri andranno in coda all'eventuale abbonamento esistente)
- tutto l'archivio delle riviste Firmware pubblicate mensilmente dal febbraio 2010

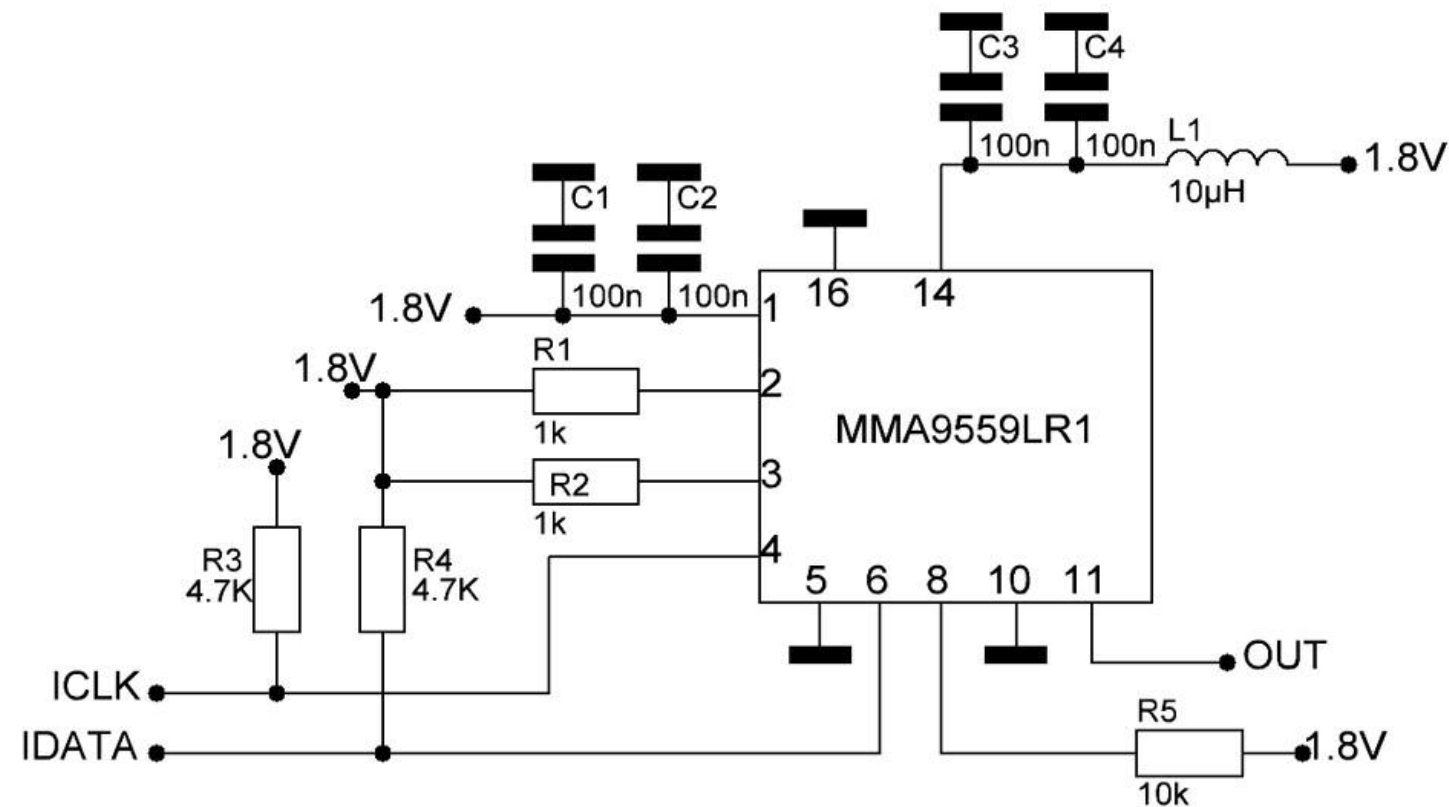
Bonus Pack Genius (solo per la card Genius Full)

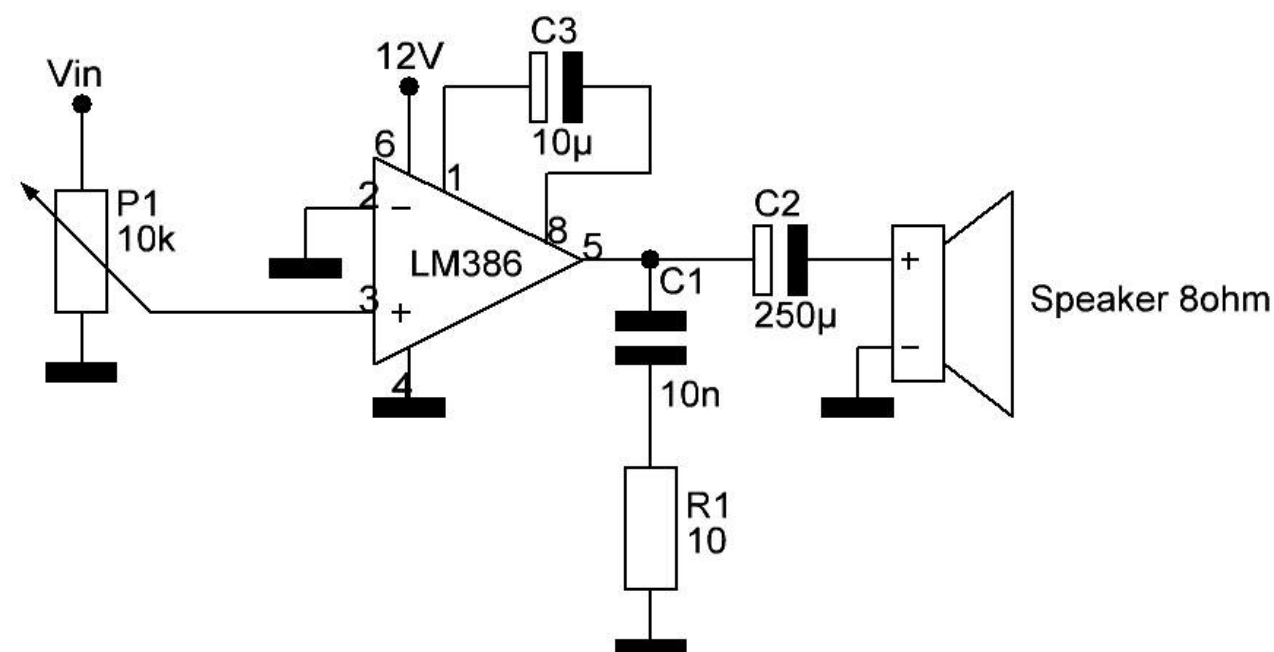
- accesso e/o download agli articoli (centinaia) del Club di Firmware
- accesso e/o download agli articoli (centinaia) del Club di Fare Elettronica
- due ebook a scelta* (Smartcard, CPLD, Display LCD, PIC, Operazionali, alimentatori, linguaggio C, Basic per PIC, Elettronica Analogica)
- due raccolte delle annate di Fare Elettronica in PDF (dal 2003 al 2012) e Firmware (dal 2006 al 2012).

SCEGLI LA TUA MEMBERSHIP E FARAI PARTE ANCHE TU DEL CLUB PIÙ ESCLUSIVO DELL'ELETTRONICA!

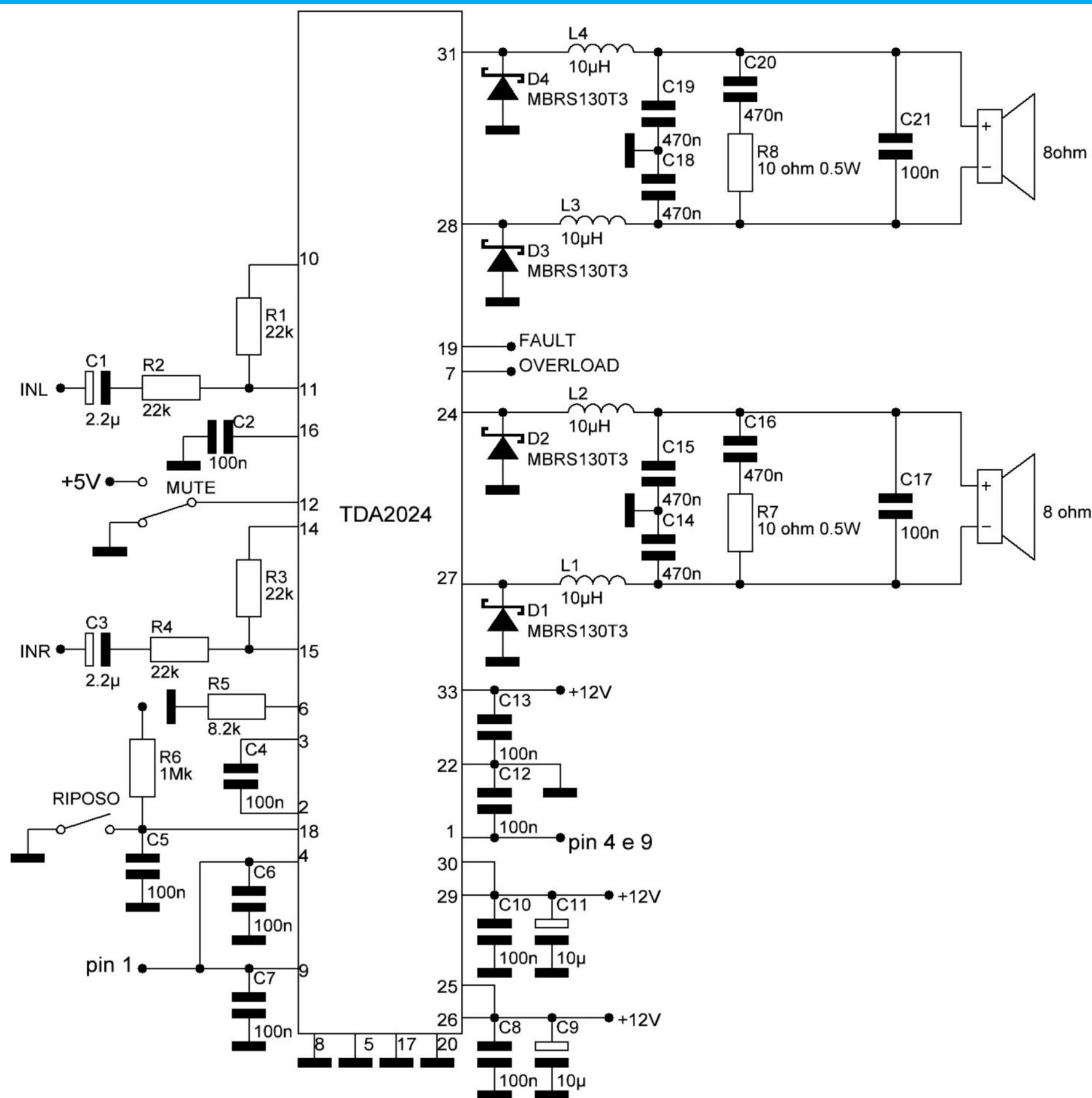
I prezzi delle card si intendono IVA inclusa

*la scelta viene effettuata al ricevimento della card

Idee di progetto

ACCELEROMETRO

Lo schema in figura è un accelerometro realizzato con un circuito integrato di Freescale. Il vantaggio di questo schema è quello di poter realizzare applicazioni basate sul riconoscimento dei movimenti , contapassi, compensazione di inclinazione. Per far funzionare questo potentissimo circuito integrato bastano solo 1.8V.

AMPLIFICATORE CLASSE AB DA 1W

L'amplificatore disegnato fornisce una potenza massima di 1Watt. Tale amplificatore può essere impiegato per amplificare segnali uscenti da una chitarra passiva o per amplificare segnali sonori uscenti da un computer. Il suo formato "DIL8" lo rende utilizzabile in qualsiasi applicazione senza dover ricorrere ad adattatore di package particolari.

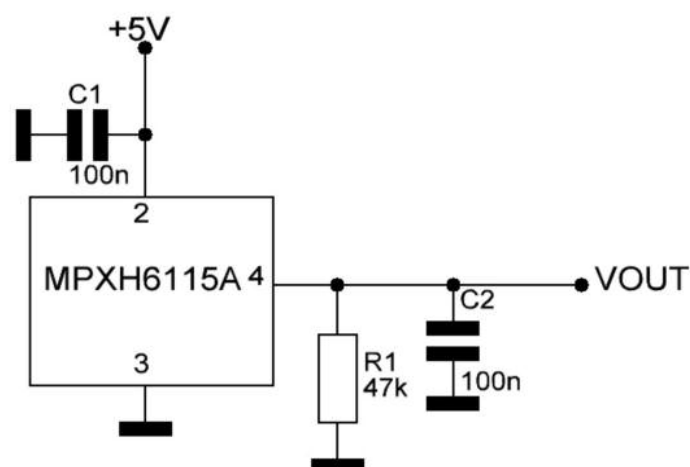
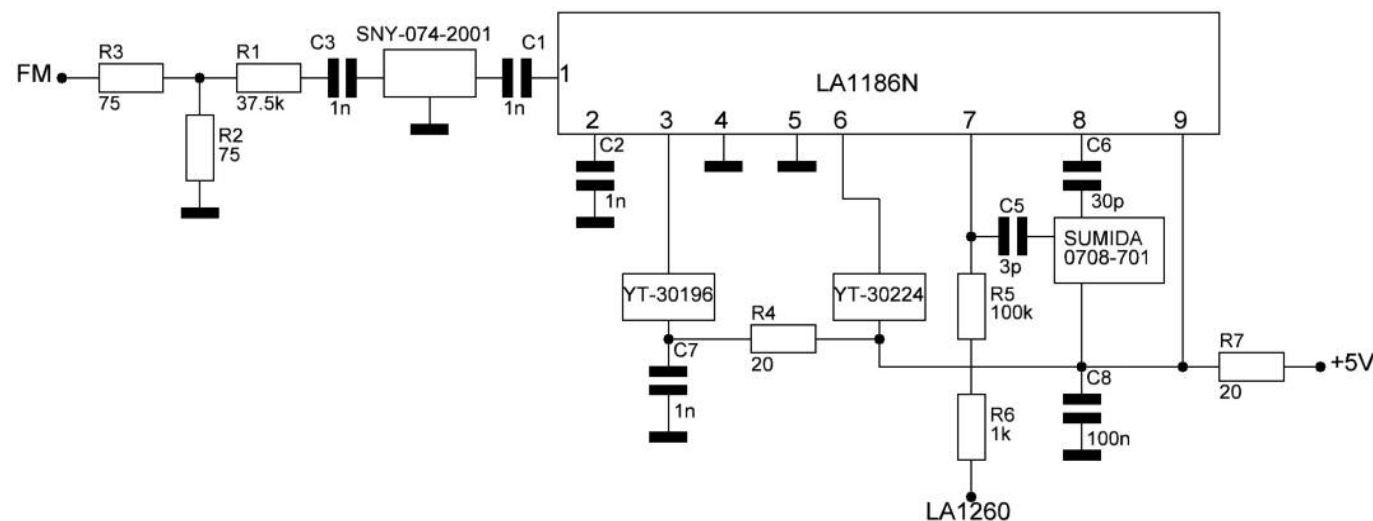
Idee di progetto

AMPLIFICATORE CLASSE T DA 10W

Nella figura è riportato un amplificatore di classe T con una potenza massima di 15Watt a canale. Questi amplificatori stanno del tutto sostituendo i vecchi amplificatore di classe AB, sia per le ridotte dimensioni sia per la dissipazione di potenza, visto che quelli di classe T hanno un rendimento prossimo a 1.

Segui le altre idee di progetto su lungocircuito blog.

Idee di progetto

MISCELATORE AUDIO PER RADIOFREQUENZA FM

In figura è riportato un miscelatore per radiofrequenza FM, utilizzato nel campo televisivo e nelle centraline per radiomicrofoni. La tensione necessaria al funzionamento può variare dai 2V agli 8V. Il vantaggio di tale circuito è quello di richiedere poche parti esterne per funzionare. Si consiglia di utilizzare i filtri con le varie sigle presenti nello schema riportato.

IGROMETRO CON MPXH6115A

L'igrometro in figura è realizzato con un MPXH6115A. Tale sensore integra al suo interno un piccolissimo circuito di compensazione della temperatura, per poter dare in uscita una tensione stabile e non variabile a sbalzi termici. La serie dei sensori MPXH sono realizzati con un sottile strato di silicio monolitico e sono del tipo piezoresistivo. Il vantaggio di essere monolitico riduce la presenza di correnti parassite e permette di avere in uscita una tensione proporzionale alla pressione. La tensione di alimentazione è di 5V.

di GIOVANNI DI MARIA

RASPBERRY CYCLETTE PI

Aggiungiamo ulteriori utili statistiche alla nostra cyclette, collegandola al Raspberry Pi

Siete stanchi delle limitate informazioni che fornisce il "computer" di bordo della vostra cyclette d camera? Con il progetto RCP (Raspberry Cyclette Pi) potrete ottenere sul televisore casalingo tante dettagliate statistiche in tempo reale e relazioni utili, ai fini di valutare correttamente la vostra performance sulla cyclette.
Si tratta di un progetto piacevole e interessante che stimolerà tutti gli amanti di questo attrezzo ginnico e gli sportivi in genere, che avranno in tal modo un motivo in più per passare qualche oretta in un coinvolgente allenamento.
L'idea proposta in queste pagine non modifica affatto ne' la parte elettronica della vostra cyclette ne' tanto meno la parte meccanica.

IL RASPBERRY CYCLETTE PI

Il Raspberry Cyclette Pi è un sistema che utilizza appunto il Raspberry Pi. Il suo sco-

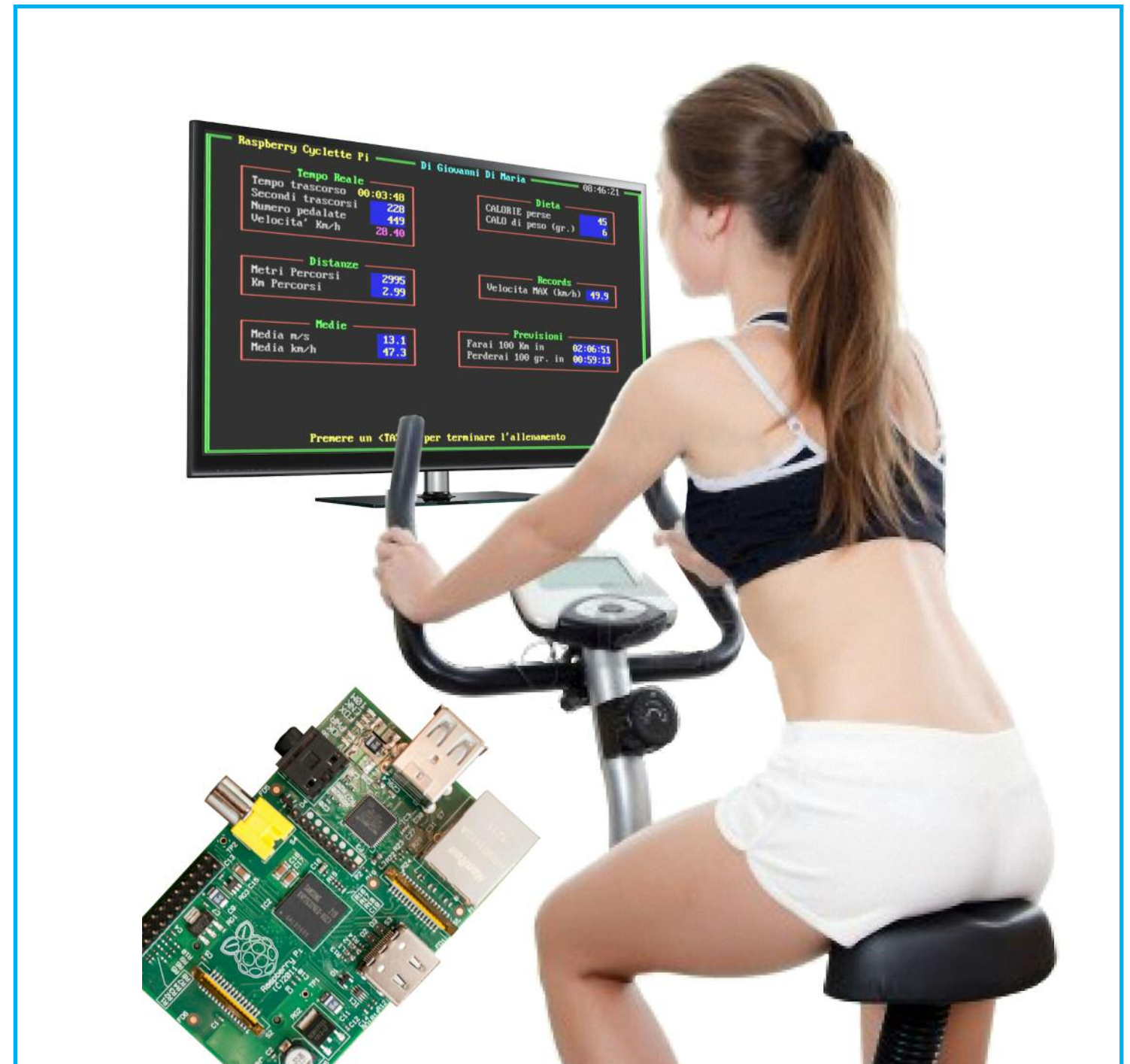



Figura 1: Raspberry Cyclette Pi

po è quello di aumentare a dismisura la qualità e la quantità di dati e misurazioni, effettuate sulla prestazione di allenamento, altrimenti impossibili da ottenere. Si basa su un sofisticato software che, in cadenza costante, calcola e visualizza una serie di importanti notizie, utili durante le pedalate.

La versione attuale del software calcola e mostra ben 14 informazioni statistiche diverse mentre, al contrario, il computer di bordo della cyclette si limita spesso a 5. Allo stato attuale esso non misura ne' fornisce la frequenza cardiaca. Ci riprometiamo di implementare tale possibilità nelle versioni future del programma.

IL COMPUTER DI BORDO DI UNA CYCLETTE

Tutte le cyclette montano a bordo un piccolo computer che fornisce, a chi si allena, utili informazioni sulla prestazione sportiva che si sta praticando. In tempo reale, infatti, un semplice display monocromatico mostra continuamente i dati rilevati.

Solitamente, le indicazioni fornite da questi dispositivi, sono le seguenti:

- Tempo: indica il tempo di utilizzo dell'attrezzo, solitamente espressa nel formato mm:ss;
- Velocità: indica la velocità attuale istantanea;
- Distanza: indica la distanza percorsa, solitamente espressa in chilometri;
- Calorie: indica le calorie bruciate durante la prestazione;
- Battiti cardiaci: indica il numero battiti cardiaci rilevati tramite un sensore posto sul manubrio, a contatto con le mani. Solitamente è espresso in bpm (battiti per minuto).

I modelli più sofisticati (e più costosi) dispongono anche di un display molto complesso, che visualizza altre tipologie di informazioni e di statistiche. Si tratta di cyclette molto professionali, spesso utilizzate nelle palestre, il cui presto farebbe impallidire chiunque.

Il display LCD che si trova nelle cyclette di fascia media visualizza, dunque, le informazioni sopra esaminate ma si limita a mostrarle solamente una alla volta, in cadenza costante temporizzata (ad esempio ogni 5 secondi). Le dimensioni molto contenute del display non permettono, infatti, grandi videate ma circoscrivono e limitano il proprio output a cinque o sei caratteri al massimo.

L'utente che trascorre un'oretta a pedalare, dunque, non è certamente divertito o interessato da tale noiosa "interfaccia" dell'attrezzo che, con buona probabilità, finisce di annoiarlo e a stancarlo prima del tempo. Questa è una delle tante cause che inducono lo sportivo ad abbandonare la cyclette da camera.

Figura 2: I connettori provenienti dai sensori

COME FUNZIONA LA RILEVAZIONE DATI IN UNA CYCLETTE TRADIZIONALE?

Come funzionano, normalmente, i computer montati a bordo delle cyclette? E come riescono a visualizzare tempi, battiti cardiaci, calorie e distanze percorse?
Come al solito sono l'elettronica e l'informatica che, a braccetto, riescono egregiamente a risolvere il problema, in maniera straordinaria.
Riducendo infatti al minimo il sistema, una cyclette non è altro che formata da due unità base:

- Un sensore di pedalata, che fornisce una sorta di base dei tempi variabile, imposta da chi pedala, da considerarsi come punto di riferimento assoluto;
- Un software (nel caso della cyclette, un firmware) che esamina gli impulsi del sensore e, processandoli, attiva tantissimi calcoli matematici e proporzioni atte a visualizzare i risultati finali.

Figura 3: Un tradizionale computer da cyclette

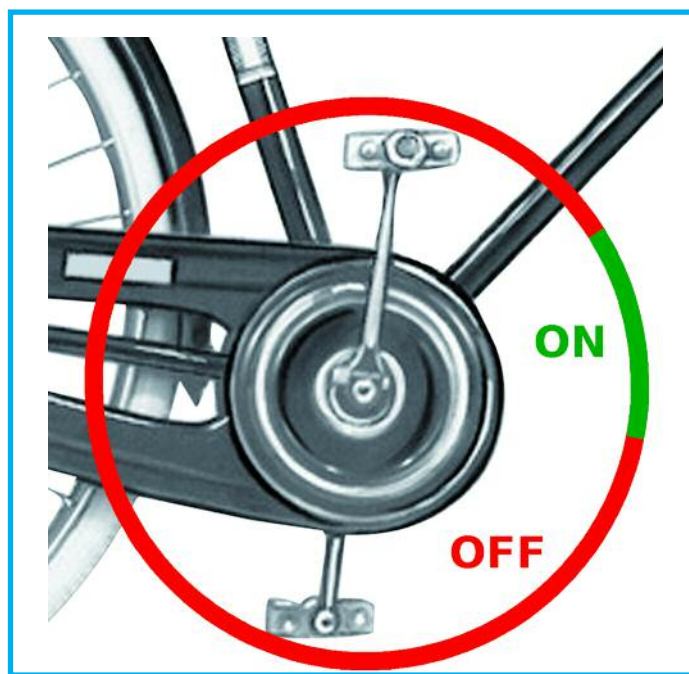



Figura 4: I due momenti di attivazione del sensore di pedalata

Il sensore

Più che sensore si tratta di un contatto (reed o altro tipo) che si attiva per un istante ad ogni giro dei pedali. I due terminali, solitamente sotto forma di piccolo spinotto jack mono, durante un'ampiezza angolare di pedalata di almeno 340°, presentano un'alta resistenza (infinita) mentre per una ristretta angolazione risultano in corto, o quasi (normalmente la resistenza non supera i 2 ohm). Questo sistema si comporta pertanto da ottimo interruttore ON/OFF e offre un buon supporto anti rimbalzo.
E' facile rintracciare tale connettore. Si tratta di staccare, dal computer a bordo della cyclette, i due spinotti jack. Uno fa capo proprio a questo interruttore di pedalata, l'altro è collegato al sensore cardio, quest'ultimo abbastanza complesso e delicato.
Con un tester, in posizione ohmetro o prova di continuità, collegato ai capi dello spinotto jack, si effettui la misurazione, pedalando molto lentamente. Ogni giro di pedali corrispondente ad un breve stato di "cortocircuito", o ad un beep del tester, testimonia che il connettore è stato indovinato.

Il Firmware

Solitamente i computer di bordo di una cyclette è dotato di un microcomtrollore. Esso si occupa di tutta la gestione di rilevazione dell'impulso proveniente dai pedali, della complessa e impegnativa elaborazione matematica di tante formule, con una lunga serie di proporzioni in virgola mobile e, infine, della rappresentazione dei dati, opportunamente cadenzata, sul display.
Anche se, durante l'uso di una cyclette, possa impressionare la semplicità del sistema, assicuriamo che "dietro le quinte" esiste una procedura abbastanza complessa, che ha messo sicuramente a dura prova i programmatori e i progettisti.

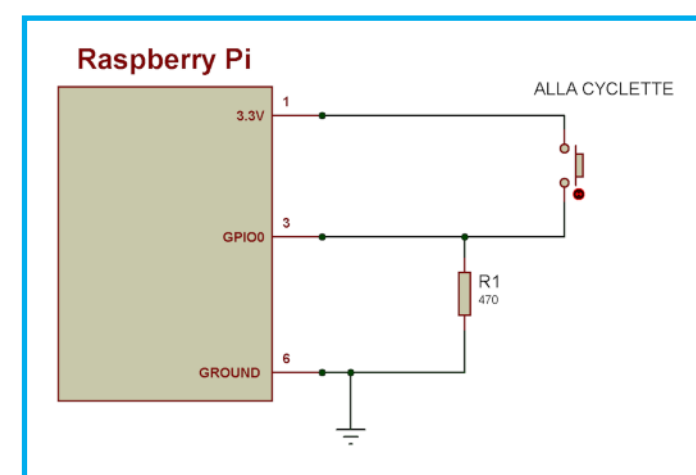



Figura 6: Schema elettrico

FILOSOFIA DEL PROGETTO

Il funzionamento del progetto è, tutto sommato, relativamente semplice. Lo spinotto jack, proveniente dalla cyclette, fornisce, ad ogni giro completo dei pedali, un impulso "passivo".

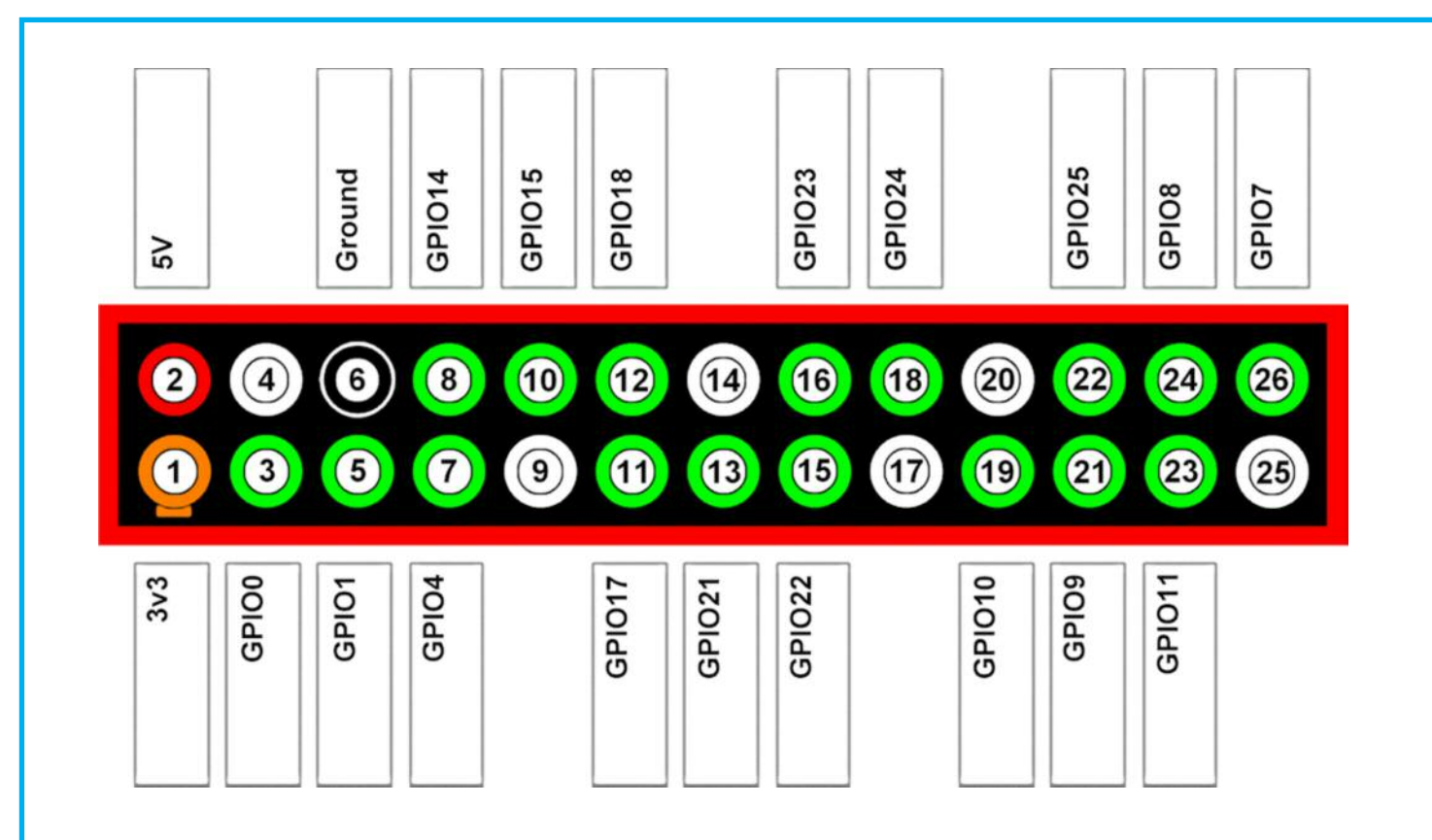



Figura 5: La porta GPIO del Raspberry Pi

Ciò vuol dire che per quasi tutta la pedalata, i due contatti elettrici sono aperti mentre, per una breve porzione di giro dei pedali, i due contatti si trovano in corto circuito, ossia collegati elettricamente tra loro.
Si tratta, in pratica, di un normale interruttore on/off, che può essere sfruttato con successo nel circuito.
Una porta del Raspberry Pi, configurata come ingresso digitale, riceverà la successione degli impulsi delle pedalate e da esse si riuscirà a calcolare innumerevoli informazioni, grazie al software implementato nel sistema.

SCHEMA ELETTRICO

Chiamare schema elettrico il sistema realizzato è alquanto esagerato. Si tratta infatti di una porta del Raspberry Pi, programmata per funzionare da ingresso digitale. Esso è collegata a massa attraverso una resistenza di pull-down da 470 ohm. Se la pedalata non attiva il contatto, tale porta si trova su un livello logico basso. Appena i pedali attivano il contatto all'interno della stretta zona di attivazione, l'interruttore si chiude e permette il passaggio della tensione di 3.3V, portando lo stato logico della porta ad uno stato alto. Il programma eseguibile farà tutto il resto.
La porta del Raspberry Pi utilizzata è la GPIO0. Durante il breve lasso di tempo, durante il quale la tensione fluisce attraverso il contatto (di parla anche di pochi centesimi di secondo), la resistenza di pull-down è attraversata da una corrente pari a circa 7mA, molto distanti dalla soglia massima prevista di 50mA. Per maggiori informazioni su come programmare e utilizzare gli ingressi del Raspberry Pi, si consultino i precedenti articoli di Fare Elettronica, sempre dello stesso autore.

Figura 7: Il cablaggio del sistema

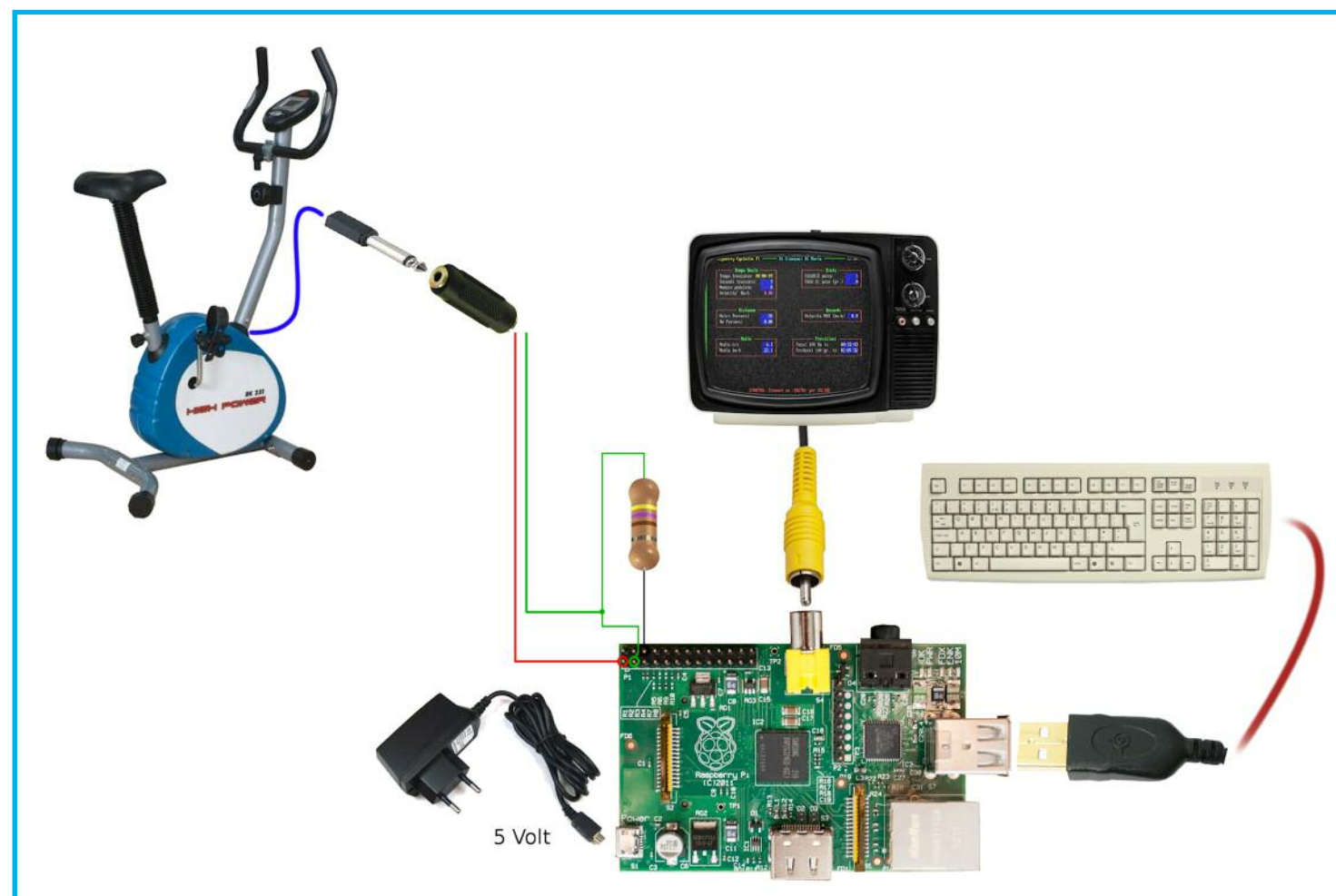
IL CABLAGGIO

Il cablaggio del sistema è molto semplice, ma occorre prestare le dovute attenzioni, onde salvaguardare la vita del Raspberry Pi. Un errore accidentale, infatti, potrebbe compromettere per sempre la vita del vostro pico computer.
La prima operazione è quella di intercettare e collegare al sistema lo spinotto jack della cyclette che fa capo al suo computer di bordo, con i metodi spiegati prima. Quindi si può passare a connettere una tastiera USB nell'apposito alloggiamento e, se si lo desidera, anche un mouse dello stesso tipo, sebbene ai fini dell'applicazione, esso risulta inutile.
Si colleghi quindi un televisore alla presa RCA (di colore giallo) o tramite la linea HDMI, la quale ultima assicura una qualità di visione eccezionale.
Si passi adesso al collegamento di una resistenza da 470 ohm. Un reoforo va collegato a massa, sul pin 6 del Raspberry Pi, mentre l'altro lato fa capo alla porta GPIO0, corrispondente al pin 3. Questo assicura uno stato logico basso alla porta, in assenza di segnali esterni.
Quindi si colleghino i due terminali provenienti dalla cyclette, rispettivamente al pin 1 e al pin 2 del Raspberry Pi. L'orientamento dei fili non ha rilevanza, dal momento che i cavetti della cyclette non fanno altro che chiudere o aprire un contatto elettrico.
Se tutto è a posto, è possibile fornire tensione al circuito e sintonizzare il TV sul canale AV o su quello adibito alla ricezione dei segnali esterni, attraverso la porta HD. Si dovrebbe vedere immediatamente l'immagine del sistema operativo, che esegue il boot.

PCB

Data la grande semplicità del circuito non abbiamo realizzato un circuito stampato. L'utente, se lo desidera, può comunque approntare una basetta per ospitare la resistenza e i connettori. Conviene realizzare il PCB sia per un motivo prettamente estetico che per la sicurezza dei contatti elettrici. Sarebbe opportuno, in aggiunta, reperire o realizzare anche un contenitore per il Raspberry Pi. In una delle prossime puntate tratteremo proprio questo tema.

AVVIARE IL PROGRAMMA

Bene, siamo arrivati al momento della prova e del collaudo del sistema. La prima cosa da fare è quella di scaricare il programma applicativo dal sito di Fare Elettronica oppure all'indirizzo http://www.elektrosoft.it/download.asp.
Si tratta di un software che gestisce totalmente la connessione con la cyclette ed il calcolo e la visualizzazione dei computi statistici.
Esso lo si può scaricare direttamente, utilizzando il Raspberry Pi, se esso è connesso ad Internet, oppure lo si può memorizzare su un PC e poi, con una pendrive USB trasportarlo sul sistema.
Il file è un eseguibile Linux, pertanto occorre accertarsi che abbia i permessi di esecuzione ("x"). Per permettere il corretto

funzionamento e l'utilizzo delle porte di comunicazione, l'utente deve "loggarsi" come "root" durante la sessione operativa, pena il non funzionamento della ricezione dei dati.
Il programma può essere memorizzato in una qualsiasi cartella del file system, consigliamo comunque di copiarlo in /home/pi oppure in /usr/bin.
Per far partire il programma basta inoltrare il seguente comando da prompt:

```
./cyclette
```

Se l'eseguibile è memorizzato in /usr/bin non è necessario di inserire "./" nel comando, dal momento che tale directory è mappata come percorso di ricerca di default, nella gestione del PATH.

INSERIRE I METRI PER PEDALATA

Dal momento che il programma può funzionare con qualsiasi tipi di cyclette da camera, occorre immettere all'inizio della sessione del software un parametro molto importante: la distanza percorsa per ogni pedalata, espressa in metri.
Infatti, nel mercato esistono tantissimi modelli dell'attrezzo, con la conseguenza che una pedalata di una cyclette, probabilmente non corrisponderà ad un'altra pedalata di un prototipo diverso, in termini di distanza percorsa ed estensione metrica lineare.
E' molto semplice misurare l'estensione di una pedalata, basta seguire i seguenti passi:

- Collegate normalmente il computer di bordo alla sua cyclette. Il Raspberry per il momento è scollegato;
- Azzerate tutti i dati;
- Eseguite 100 pedalate (o ancora meglio, 1000, per una maggior precisione);
- Dividete la distanza percorsa per 100 (o per 1000). Il risultato finale esprime la strada percorsa con una sola pedalata. Ricordiamo che il valore deve essere espresso in metri.

Figura 9: La videata del software che misura tante informazioni sulla prestazione

Figura 8: La richiesta della distanza percorsa con una pedalata

E ADESSO... PEDALIAMO.

Dopo aver confermato la distanza che si percorre, con una sola pedalata, il programma invita a premere un tasto, quando si è pronti a iniziare la sessione ginnica.
Si salga pertanto sulla cyclette, si prema un tasto qualsiasi della tastiera e si inizi a pedalare. Immediatamente una maschera molto dettagliata inizia a mostrare sul video tanti dati e tante informazioni che si aggiornano ad ogni secondo.
Quando si intende concludere la sessione di allenamento, si può terminare di pedalare e, allo stesso tempo, bisogna premere qualsiasi taso della tastiera collegata al Raspberry Pi. In questo modo le informazioni finali non sono falsate dai tempi morti di non pedalata.

LE STATISTICHE AGGIUNTIVE IN DETTAGLIO

Il sistema proposto in queste pagine propone allo sportivo tante utili statistiche in più, che non si potrebbero mai avere a disposizione e commentare con una normale cyclette di fascia economica. Il software propone una classificazione delle informazioni suddivise in sei categorie: titolo, tempo reale, distanze, medie, dieta, records, previsioni. Ecco, in dettaglio, i dati forniti:

- **Titolo**
  - Nome programma;
  - Nome autore;
  - Orologio;

- **Tempo Reale**
  - Tempo trascorso dall'inizio della prestazione;
  - Secondi trascorsi dall'inizio della prestazione;
  - Numero delle pedalate effettuate;
  - Velocità istantanea in km/h;

- **Distanze**
  - Metri percorsi;
  - Chilometri percorsi;

- **Medie**
  - Media espressa in metri al secondo;
  - Media espressa in chilometri all'ora;

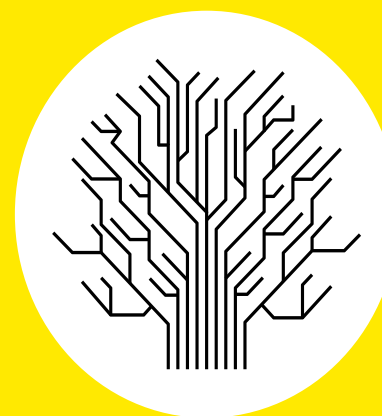
- **Dieta**
  - Calorie perse durante la pedalata;
  - Calo di peso, espresso in grammi;

- **Records**
  - Velocità massima raggiunta nella sessione corrente;

- **Previsioni**
  - Tempo totale previsto per percorrere 100 chilometri teorici;
  - Tempo totale previsto per perdere il peso di 100 grammi teorici.

Come si vede dunque si hanno a disposizione tanti dati statistici con i quali si possono valutare le proprie prestazioni ginniche con maggiore qualità ed efficacia.

CONCLUSIONI

Il Raspberry Pi dà tanti spunti e idee che nemmeno l'immaginazione riesce ad elencare. Con esso, tutti i progetti sono effettivamente realizzabili, a differenza dei microcontrollori, che hanno alcuni limiti. Pensiamo, ad esempio, alla possibilità di vedere i risultati su una televisione oppure di memorizzare i dati su un grande supporto di massa; queste sono tutte potenzialità estremamente preziose che, adesso, il programmatore ha facilmente a portata di mano.
Con il sistema proposto in queste pagine, assicuriamo che fare ginnastica è sicuramente meno faticoso. Naturalmente è solo una questione psicologica, che si basa sulla differente metodologia di rappresentazione dei dati e delle informazioni. Esse sono più ricche, più numerose e, sicuramente, meno noiose di quelle visualizzate da un monotono e incolore monitor da cyclette. Durante le prove delle pedalate si riscontrerà uno spirito agonistico maggiore e un relativo divertimento nell'osservare, a colori, le proprie prestazioni, espresse in tantissime grandezze diverse. La valutazione della prestazione avrà più angolazioni di analisi.
Una versione futura del progetto prevedrà anche la visualizzazione cardio e le statistiche ad essa connessa, senza apportare alcuna modifica alla cyclette.
Il progetto può essere utilmente adoperato anche con i tapis roulant magnetici, che utilizzano la medesima metodologia elettrica. In questo caso occorre fornire al software la distanza percorsa ad ogni giro completo del tappeto.
Il calcolo delle calorie è invece indicativo e si basa sui calcoli matematici effettuati dai normali computer da cyclette in commercio. I risultati infatti sono puramente orientativi e non devono essere presi per dati assoluti, ma costituiscono comunque un buon punto di riferimento.

Elenco componenti

	Raspberry Pi, mod. A o B
R1	Resistenza 470 ohm ¼ W
	Televisore o monitor con presa RCA o HDMI
	1 Tastiera per PC USB
	Alimentatore 5V - 700mA
	Cavetterie varie
	Presa volante jack femmina piccola mono
	Cyclette da camera

di ANGELO DE BARTOLO

ESPANDERE LE PORTE DI I/O CON PCA8574

Quando si utilizzano i microcontrollori nei sistemi elettronici spesso accade di rimanere a corto di porte di INPUT / OUTPUT a causa di dispositivi "esigenti" quanto a numero di porte (es: display LCD, tastiere, pulsanti). Per sopperire a tale necessità è possibile utilizzare un circuito integrato come il PCA8574 che aggiunge 8 porte quasi bidirezionali ai nostri microcontrollori

Il PCA8574 è un *port expander* che viene istruito tramite protocollo di trasmissione I2C a velocità massima di 400kHz: in questa sede verrà fatto riferimento allo specifico integrato prodotto da NXP (già Philips Semiconductors).
Il dispositivo (schema a blocchi in figura 1) è provvisto di 8 porte quasi-bidirezionali, un interfaccia per il bus I²C completo di 3 bit di indirizzamento ed un pin di interrupt. L'indirizzo è a 7 bit di cui i primi 4 bit sono impostati fissi al valore "0 1 0 0" mentre gli altri 3 bit sono impostabili tramite un semplice ponticello verso massa (con relativa resistenza di pull-down) o verso l'alimentazione positiva rendendo possibile quindi l'utilizzo di 2^3 = 8 dispositivi del tipo PCA8574. C'è da dire che la NXP commercializza un dispositivo identico ma con una lettera "A" come suffisso (PCA8574A) la quale ha come indirizzo fisso "0 1 1 1" aumentando così il numero di *port expander* che si possono collegare allo stesso bus I²C.

PORTE QUASI-BIDIREZIONALI

Una porta si definisce quasi bidirezionale quando può essere utilizzata sia come In-

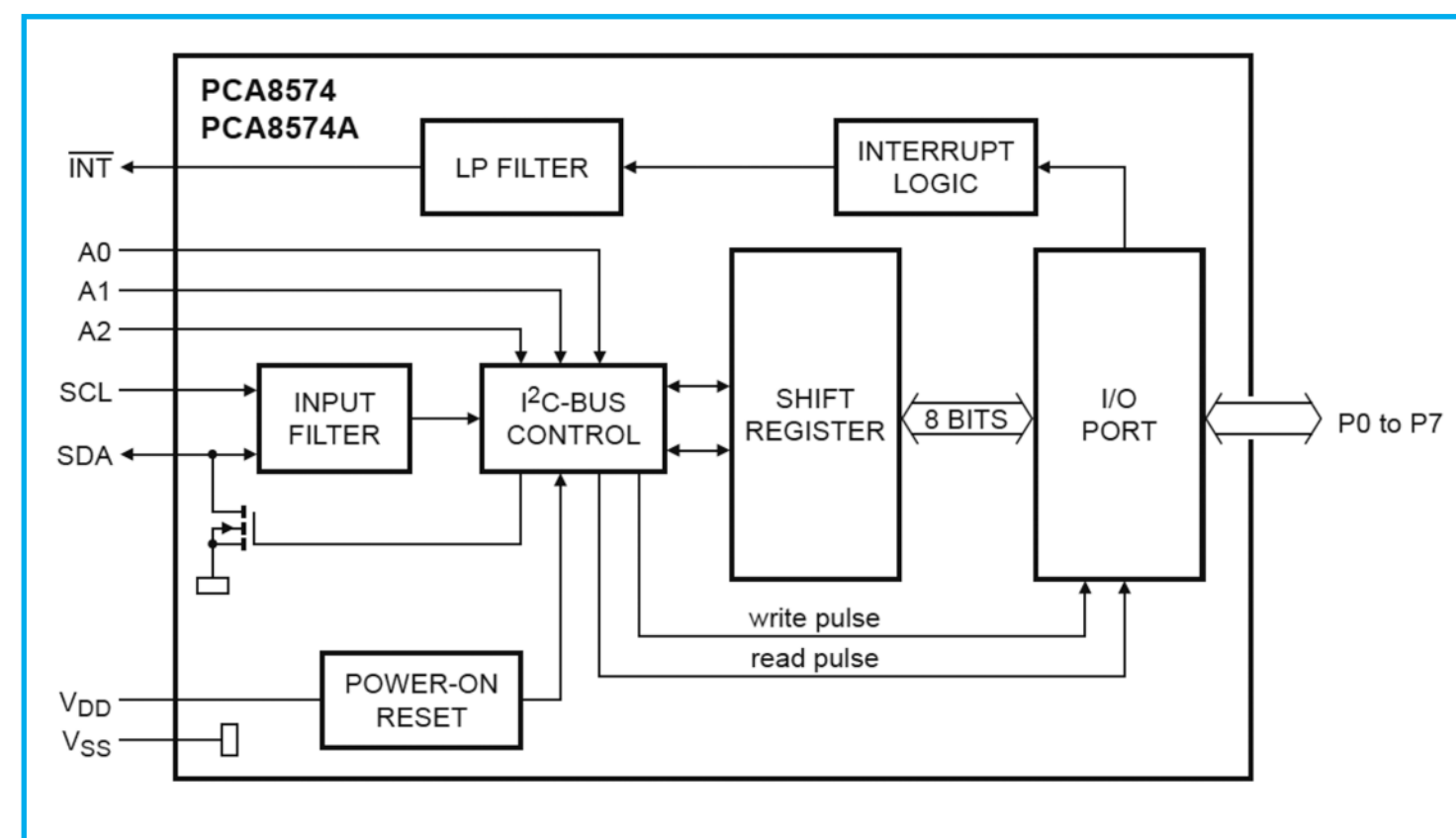



Figura 1: Schema a blocchi

put che come Output senza l'utilizzo di un segnale di controllo o di un settaggio del registro. Ogni qual volta si vuole utilizzare una porta come INPUT bisogna impostarla in uno stato logico HIGH: si osserverà una debole corrente di 100µA dovuta ad un sistema di pull-up interno (essenzialmente un P-channel). La porta potrà cambiare lo stato logico in LOW sia tramite un transistor (N-channel) interno (usandola quindi come porta di OUTPUT) che tramite un segnale esterno (utilizzandola quindi come porta di INPUT) come evidenziato nella figura 2.
All'avvio il POR (Power-On-Reset) inizializza le porte come INPUT e si osserverà, pertanto, un livello logico HIGH.
Bisogna porre attenzione al limite di corrente erogabile da ciascuna porta: in uno stato di LOW OUTPUT la singola porta può erogare al massimo 40mA (200mA in totale tra tutte le porte nello stato LOW) mentre nello stato di HIGH la corrente erogabile scende a 300µA massimi a porta. E' quindi consigliabile, o meglio indispensabile, pilotare carichi con l'uscita bassa e se necessario si possono connettere più porte in parallelo (fino ad 8 porte) in modo da aumentare la corrente di uscita totale (figura 3) che, come già detto, potrà essere al massimo di 200mA e sarà limitata con eventuali resistenze.

UTILIZZO DELLE PORTE DI I/O: SCRITTURA

L'operazione di scrittura di un livello logico sulle porte si compie in perfetto standard I2C pertanto inizia con un segnale di Start da parte del Master, seguito da un byte composto dall'indirizzo dello Slave (il nostro PCA8574) e dal bit R/W = 0 (che indica appunto un'operazione di scrittura) ed infine da uno o più Data-Byte contenenti i valori dei bit corrispondenti alle porte a partire dal bit più significativo (il primo bit imposterà la porta P7) finendo al bit meno significativo (l'ultimo bit andrà ad agire sulla porta P0). Si ribadisce quanto detto in precedenza: volendo utilizzare una porta come INPUT è necessario impostarla al valore logico HIGH. In figura 4 è rappresentato un tipico ciclo di scrittura del dispositivo in cui la porta P5 viene settata ad un valore HIGH (con il Data-Byte-1) e pertanto tale porta potrà essere utilizzata anche come porta di INPUT oltre che a rappresentare lo stato logico di OUTPUT.

Figura 3: Metodo di pilotaggio dei carichi

Figura 2: Schema delle porte quasi-bidirezionali

UTILIZZO DELLE PORTE DI I/O: LETTURA

Anche la lettura del valore delle porte inizia con un segnale di Start da parte del Master, seguito dall'Address-Byte del dispositivo Slave più l'ultimo bit di R/W =1 (che indica un'operazione di lettura). A questo punto il dispositivo PCA8574 che era uno *Slave-Receiver* ora diventa uno *Slave-Transmitter* e dopo aver mandato il segnale di ACK al Master inizia a trasmettere una sequenza di bit corrispondenti allo stato logico delle porte (vedi figura 5) e successivamente resetta il proprio pin di interrupt. Successivamente il master invia un segnale di NACK e di STOP terminando la lettura.

IL PIN DI INTERRUPT

Il dispositivo è provvisto di un pin di interrupt a logica negata (quando il pin si attiva si porterà allo stato logico LOW) che interviene quando uno degli ingressi cambia di stato avvertendo così il microcontrollore che sta a monte (figura 6).
Il pin si resetta - riportandosi allo stato HIGH - quando il master compie un'operazione di lettura o scrittura del dispositivo oppure quando il pin che ha azionato l'interrupt ritorna allo stato precedente.

Figura 4: Un tipico ciclo di scrittura

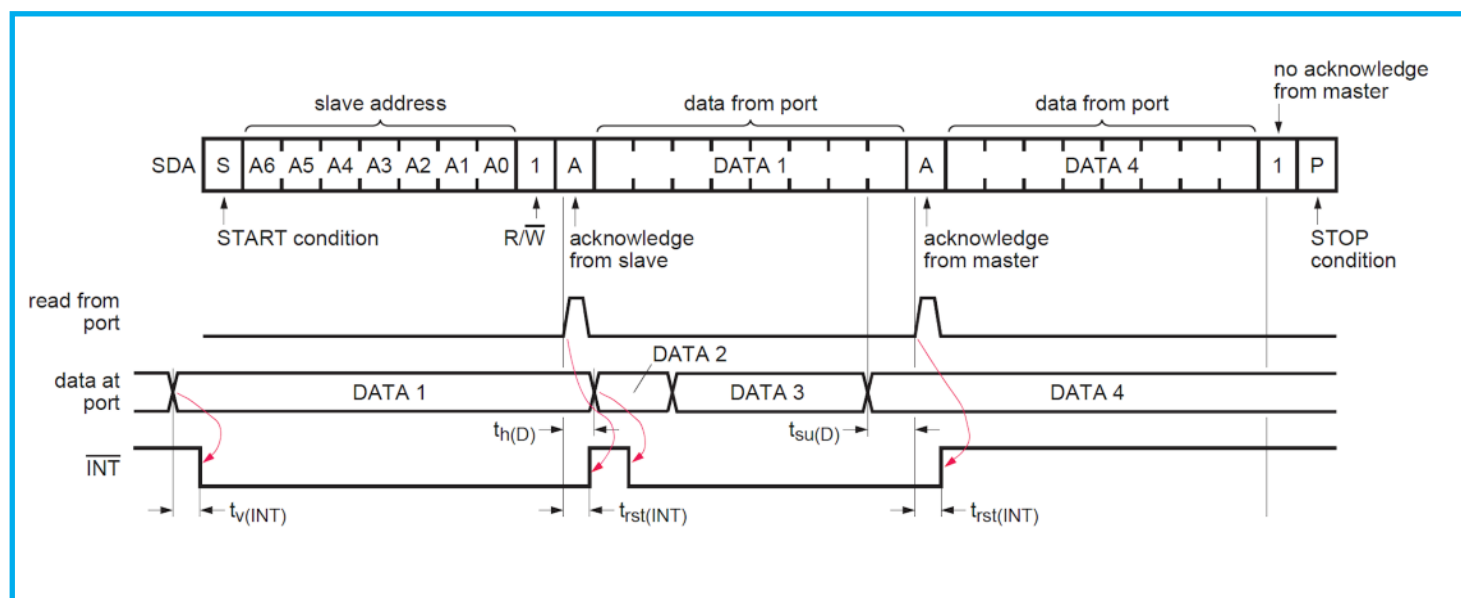



Figura 5: Un tipico ciclo di lettura

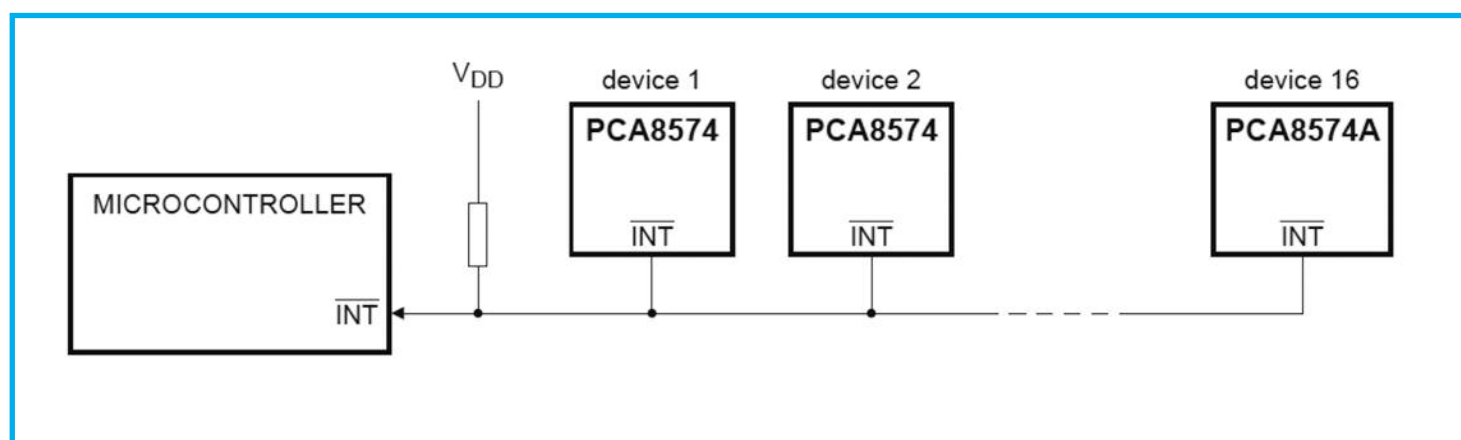



Figura 6: Linea di Interrupt

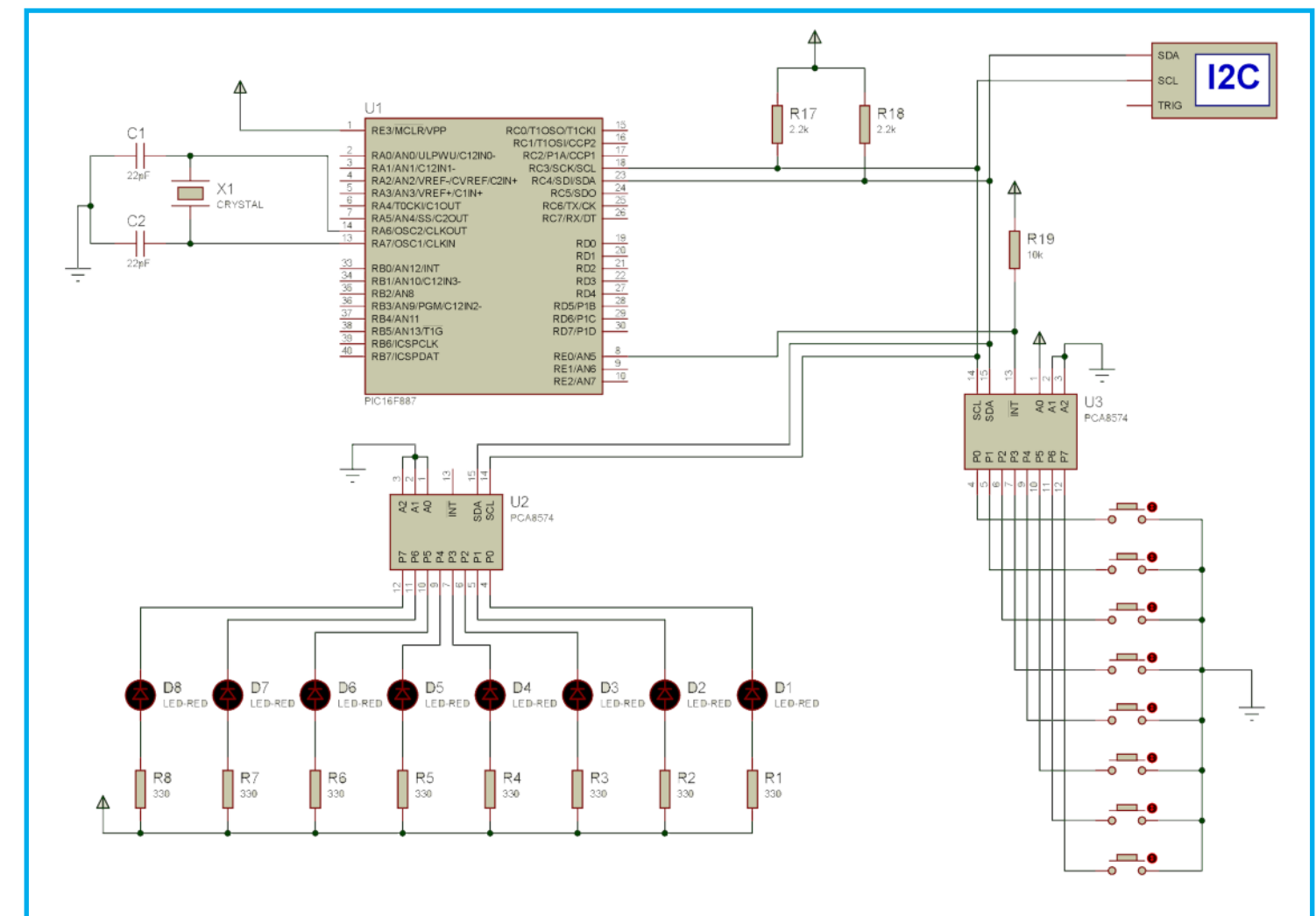



Figura 7: Schema elettrico con PCA8574 e PIC16F887

Il comportamento del segnale di interrupt durante i cicli di scrittura e lettura è evidenziato nelle figure 4 e 5.

ESEMPIO DI UTILIZZO CON PIC16F887 E MIKROC

Un semplice esempio didattico di utilizzo è mostrato nello schematico seguente (figura 7) dove sono stati utilizzati due PCA8574 di cui uno utilizzato come ingresso per ricevere segnali da push-button ed uno utilizzato per espandere le uscite con dei led. I pin di interfaccia I2C sono il pin 18 per il segnale di clock (SCL) ed il pin 23 per i dati (SDA) ed entrambi vanno impostati in uno stato di alta impedenza ovvero vanno impostati come degli INPUT (comando PORTC = 0xFF). Per realizzare il bus I2C sono stati scelti resistori di valore 2.2k (vedi box di approfondimento) mentre l'oscillatore è esterno a 8MHz con classici condensatori a 22pF. Il primo port-expander è stato settato con indirizzo (a 7 bit) 0100000 ponendo i 3 pin A0-A1-A2 a massa mentre il secondo port-expander è stato settato con indirizzo 0100001 ponendo il pin A0 a V_{DD} ed a massa i pin A1-A2. Essendo l'integrato limitato ad un massimo di 300µA come *source current* ed a 40mA di *sink current* per pin (totale dispositivo 200mA) è stata scelta (obbligata) la modalità *sink current* pertanto ogni pin di output sarà collegato a V_{DD} tramite una resistenza di 330 per ciascun led (alimentato con circa 14mA) rimanendo ampiamente sotto la soglia limite dei 200mA: ciò implica che

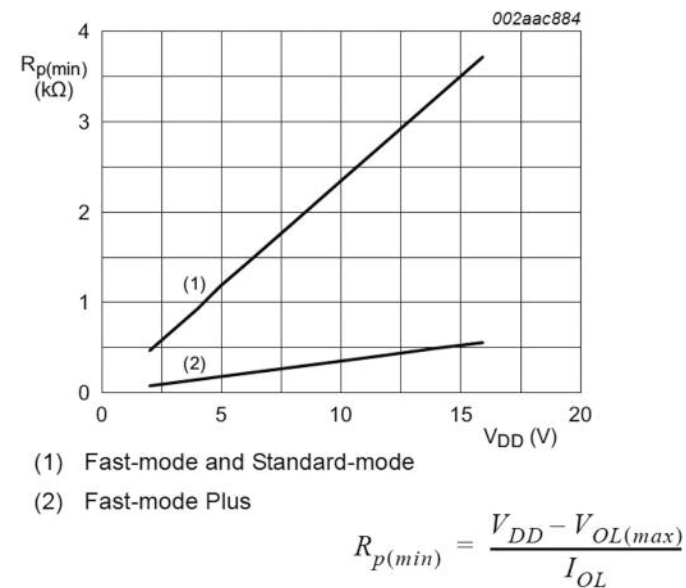
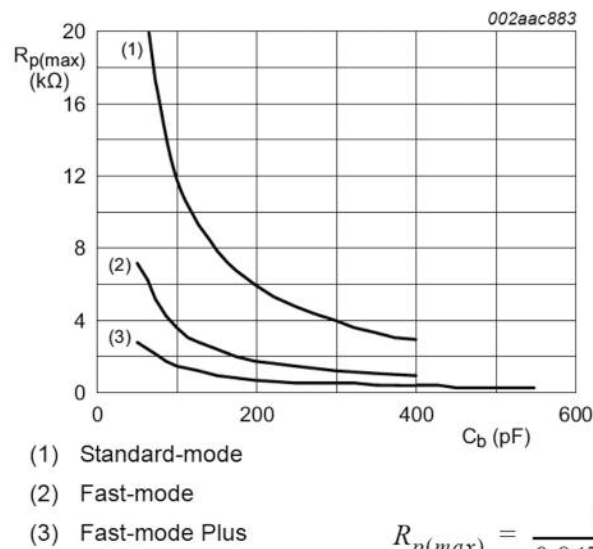



Figura 8: Valori dei resistori di pull-up nel bus I2C

per accendere un led bisogna portare l'uscita corrispondente ad un livello basso mentre se si vuole tenere spento il led bisogna mantenere alto il pin corrispondente. Sul secondo port expander - usato per estendere gli ingressi - si è collegato un pulsante verso massa (si può fare a meno del resistore di pull-up e pull-down). Il pin di interrupt è stato mantenuto alto tramite una resistenza di pull-up di 10k. Il microcontollore della MicroChip è stato opportunamente programmato in linguaggio mikroC della casa MikroElektronika il cui listato è stato integralmente riportato.

COMMENTI AL LISTATO

Il listato è composto da un loop infinito in cui viene costantemente monitorato lo stato dell'interrupt del port expander con i pulsanti sul pin di input RE0 inserendo una pausa per evitare il problema del debouncing dei pulsanti.
Quando un pulsante è premuto l'interrupt del PCA8574 si porta ad un livello basso, il dato viene letto e l'interrupt si riporterà ad

RESISTORI DI PULL-UP NEL BUS I2C

Un bus standard I2C ha come limite la capacità intrinseca delle connessioni, cavi e pin e questo valore delimita (in termini di costante di tempo RC) il valore massimo dei resistori di pull-up per permettere un rapido fronte di salita del segnale logico. Oppositamente, il valore dell'alimentazione limita il valore minimo dei resistori di pull-up in quanto a termini di corrente massima definita dallo standard I2C (3mA in Standard e Fast Mode, e 20mA in Fast Mode Plus). Nella figura 8 sono mostrati i grafici elaborati dalla NXP rappresentanti i vincoli descritti. Considerando le specifiche del bus I2C indicate nella tabella 1 e tabella 2, utilizzando un'alimentazione del bus a 5V, ipotizzando il valore della capacità del bus al massimo consentito dal protocollo (ovvero 400pF), in modalità "Standard Mode" avremmo un valore minimo di 1.53k ed un massimo di 2.95k evidenziato dai calcoli di Figura 9. Si è optato per un valore commerciale di 2.2k .

LISTATO 1

LAMPEGGIO LED IN MIKROC TRAMITE PCA8574 E PIC16F887

```
// configuration word 0x2ff1 e 0x0700 - crystal 8mhz

bit leggere;     // indica se il dato è da leggere o meno
char dati;       // rappresenta le porte dell'expander2
char dati2;      // rappresenta le porte dell'expander1

const unsigned int expander1 = 0x40; // indirizzo 0b01000000;
const unsigned int expander2 = 0x42; // indirizzo 0b01000010;

void I2C_Scrivi(unsigned int indirizzo, char dati){ // routine per la scrittura
     I2C1_Start();            // manda uno start sul bus
     I2C1_Wr(indirizzo);      // scrive l'indirizzo dello slave sul bus
     I2C1_Wr(dati);           // scrive i dati
     I2C1_Stop();             // stoppa la scrittura
}

char I2C_Leggi(unsigned int indirizzo){ // routine per la lettura
     char temp;              // variabile tempranea
     I2C1_Start();           // manda uno start sul bus
     I2C1_Wr(indirizzo);     // scrive l'indirizzo dello slave sul bus
     I2C1_Repeated_Start(); // manda uno start-repeated
     I2C1_Wr(indirizzo+1);  // si aggiunge "1" per la lettura
     temp = I2C1_Rd(0);     // si memorizza il dato in temp
     I2C1_Stop();           // stoppa la lettura
     return temp;           // restituisce il valore letto
}

void main() {
     ANSEL = 0x00;           // imposto tutti i pin in I/O digitale
     ANSELH = 0x00;
     PORTC = 0x00;           // imposto le porte come INPUT
     PORTE = 0x11;           // imposto le porte come INPUT ed OUTPUT
     I2C1_Init(100000);      // inizializzo il bus a 100kHz
     Delay_ms(100);          // attendo che le inizializzazioni abbiano effetto
     leggere = 1;            // imposto il bit che indica quando il dato è da leggere:
                            // il dato è letto quando il pulsante è premuto (non rilasciato)
                             // mentre non viene letto quando è rilasciato

     while (1) {             // ciclo infinito
           if ((PORTE.F0 == 0) && (leggere==1)) {
                                          // se l'interrupt è attivo ed è da leggere
              Delay_ms(10);               // ritardo per antidebouncing
              dati = I2C_Leggi(expander2);    // legge i dati del pulsante premuto
              dati2 = I2C_Leggi(expander1);   // memorizza l'attuale situazione dei led
              if (dati.B0 == 0) {             // verifica il pulsante associato al bit0
                 dati2.B0 = !dati2.B0;        // inverte lo stato al led
                 I2C_Scrivi(expander1, dati2); // scrive sulle porte dei led
              }
              if (dati.B1 == 0) {             // come prima ma sul bit1
                 dati2.B1 = !dati2.B1;
                 I2C_Scrivi(expander1, dati2);
              }
              if (dati.B2 == 0) {             // come prima ma sul bit2
                 dati2.B2 = !dati2.B2;
                 I2C_Scrivi(expander1, dati2);
              }
              if (dati.B3 == 0) {            // come prima ma sul bit3
                 dati2.B3 = !dati2.B3;
```

```
                I2C_Scrivi(expander1, dati2);
             }
             if (dati.B4 == 0) {             // come prima ma sul bit4
                dati2.B4 = !dati2.B4;
                I2C_Scrivi(expander1, dati2);
             }
             if (dati.B5 == 0) {             // come prima ma sul bit5
                dati2.B5 = !dati2.B5;
                I2C_Scrivi(expander1, dati2);
             }
             if (dati.B6 == 0) {             // come prima ma sul bit6
                dati2.B6 = !dati2.B6;
                I2C_Scrivi(expander1, dati2);
             }
             if (dati.B7 == 0) {             // come prima ma sul bit7
                dati2.B7 = !dati2.B7;
                I2C_Scrivi(expander1, dati2);
             }
             leggere = 0;                    // abbiamo resettato l'interrupt mentre il
                                           // pulsante era premuto, rilasciandolo l'in-
terrupt
                                             // si riattiverà ma noi non elaboreremo il
dato letto
          }
          if ((PORTE.F0 == 0) && (leggere==0)){
             Delay_ms(20);                     // ritardo per antidebouncing
            // se l'interrupt è attivo ma il dato non è da leggere perchè è il rilascio
del pulsante
             dati = I2C_Leggi(expander2);     // leggo ma non elaboro
             leggere = 1;                     // il prossimo dato è da leggere
          }
       }
}
```

un livello alto; quando il pulsante viene rilasciato il PCA8574 riconoscerà questo come un altro cambiamento e riporterà l'interrupt ad un livello basso: in questo caso verrà fatta una lettura a solo scopo di resettare l'interrupt. Per mettere in pratica questo ragionamento si è utilizzato un bit destinato a tale discriminazione, il bit *leggere*. Le routine *I2C_Scrivi* e *I2C_Leggi* sono servite per ridurre inutili ripetizioni di codice dove si passa l'indirizzo a 7 bit di scrittura e la routine di lettura setta il bit LSB in lettura (riga I2C1_Wr(indirizzo+1);).

Modalità	Time Rise massimo
Standard Mode	1000ns
Fast Mode	300ns
Fast Mode Plus	120ns

Tabella 1: valore massimo del time-rise

Modalità	Corrente minima
Standard Mode	3mA
Fast Mode	3mA
Fast Mode Plus	20mA

Tabella 1: valore minimo di corrente al livello LOW

$$R_{p(max)} = \frac{1000ns}{0.8473 \times 400pF} = \frac{1000ns}{338.92pF} = 2.950k\Omega$$

$$R_{p(min)} = \frac{5V - 0.4V}{3mA} = \frac{4.6V}{3mA} = 1.53k\Omega$$

Figura 9: Valori massimi e minimi delle resistenze di Pull-Up

di GIOVANNI DI MARIA

ATTIVAZIONE TEMPORIZZATA DI UN CARICO

Una semplice ed utile realizzazione che alla pressione di un pulsante attiva un carico e lo disattiva dopo alcuni secondi

Tempo fa un cliente chiese la realizzazione di un dispositivo alquanto semplice. Doveva accendere una resistenza elettrica riscaldatrice (di potenza) per 5 secondi, alla pressione di un pulsante. La resistenza aveva il compito di fondere un sottile foglio di plastica e non doveva restare accesa per tanto tempo. Le soluzioni possibili erano molteplici: si poteva usare una porta CMOS ad alta impedenza, sfruttando la costante di tempo di un circuito RC, oppure ancora si poteva impiegare un microcontrollore programmato. Alla fine si è deciso di utilizzare l'integrato NE555 in configurazione monostabile, soprattutto per approfondire il suo aspetto didattico.

Figura 1: Operazioni del monostabile

L'IDEA

Come detto prima, il cliente doveva utilizzare di una resistenza elettrica "da taglio" con la quale poteva tagliare, grazie alla sua elevata temperatura, dei sottili fogli di plastica. L'alimentazione elettrica doveva essere erogata per 5 secondi, pena la non riuscita dell'operazione. In altri termini, la temperatura doveva risultare ottimale e ciò si poteva ottenere calibrando perfettamente il tempo di mantenimento della tensione.

A grandi linee, il dispositivo doveva avere il seguente funzionamento:

1) Inizialmente la resistenza doveva essere spenta, quindi fredda;

2) Alla pressione di un pulsante, la resistenza si riscaldava;

3) La resistenza si manteneva alimentata per 5 secondi;

4) Dopo 5 secondi la resistenza veniva scollegata dall'alimentazione;

5) Il processo si poteva ripetere ad ogni pressione del pulsante.
Ovviamente il tempo di alimentazione poteva essere regolato a piacimento.

IL CIRCUITO MONOSTABILE

Una delle migliori soluzioni è offerta dall'integrato 555, che prevede perfettamente questa casistica di problematica. Un'applicazione che implementa l'attivazione temporizzata di un carico prende il nome di circuito monostabile.
Si tratta di un circuito che presenta una tensione di uscita stabile che può essere modificata grazie a un impulso esterno che lo porta ad un altro stato. Dopo un intervallo di tempo, dipendente dai componenti esterni utilizzati, la tensione di uscita *Vout* ritorna nella condizione iniziale stabile, fino ad un altro eventuale impulso di comando esterno.
In altre parole, Il circuito è caratterizzato inizialmente da uno stato stabile. Una sollecitazione esterna lo commuta ad un altro stato, che viene mantenuto per un determinato periodo di tempo *T*. Alla fine ritorna automaticamente allo stato stabile iniziale. Un nuovo impulso esterno ripete il processo all'infinito. Un esempio di applicazione è rappresentata dalle luci di una scala in un condominio.

IL 555 COME MULTIVIBRATORE MONOSTABILE

In questo progetto, l'integrato 555 è utilizzato come multivibratore monostabile. Come detto prima, il circuito, a regime, si trova in uno stato stabile. Per rimuoverlo da tale stato e portarlo nello stato di "quasi stabile" si deve applicare un segnale esterno, detto "trigger". Lo stato di "quasi stabile" perdura per un tempo definito solitamente dai valori di una resistenza e un condensatore. Il monostabile è un temporizzatore, di solito usato per attivare un processo per un tempo ben determinato.

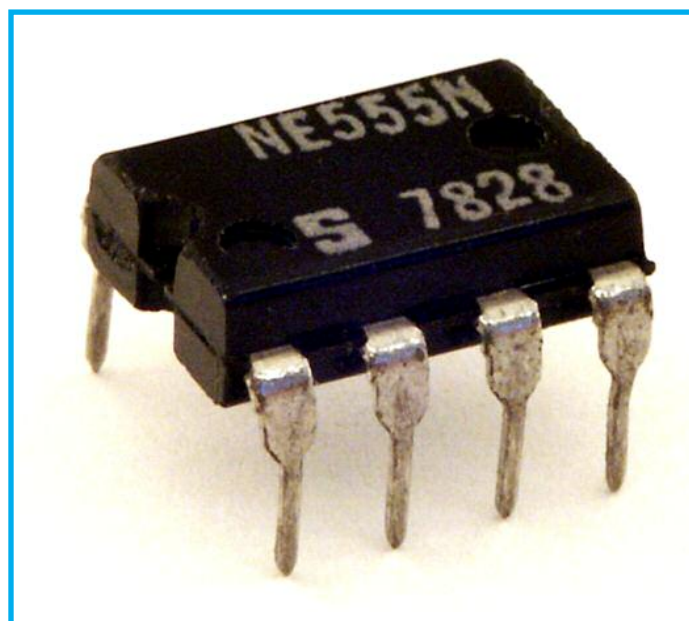



Figura 2: L'integrato NE555

SCHEMA ELETTRICO

Il circuito è alimentato dalla tensione di 12V. L'integrato NE555 è configurato in modalità di multivibratore monostabile. In fase di "riposo" il piedino 2 dell'integrato (trigger) è alto, grazie alla resistenza R1 di 10k, collegata direttamente nel ramo di alimentazione.
Anche il nodo tra R2 e C2 si trova a potenziale basso, per cui l'uscita (piedino 3) è anch'essa bassa ed il relè non risulta eccitato.
Al momento della pressione del pulsante, l'impulso di trigger attiva la temporizzazione, l'uscita si porta a livello alto, attivando il relè; il condensatore C2 comincia a caricarsi, sino a che raggiunge la tensione di circa 2/3 VCC (nel nostro caso circa 8V). A questo punto il circuito ritorna in stato di riposo, il relè si diseccita ed il sistema ritorna in attesa.

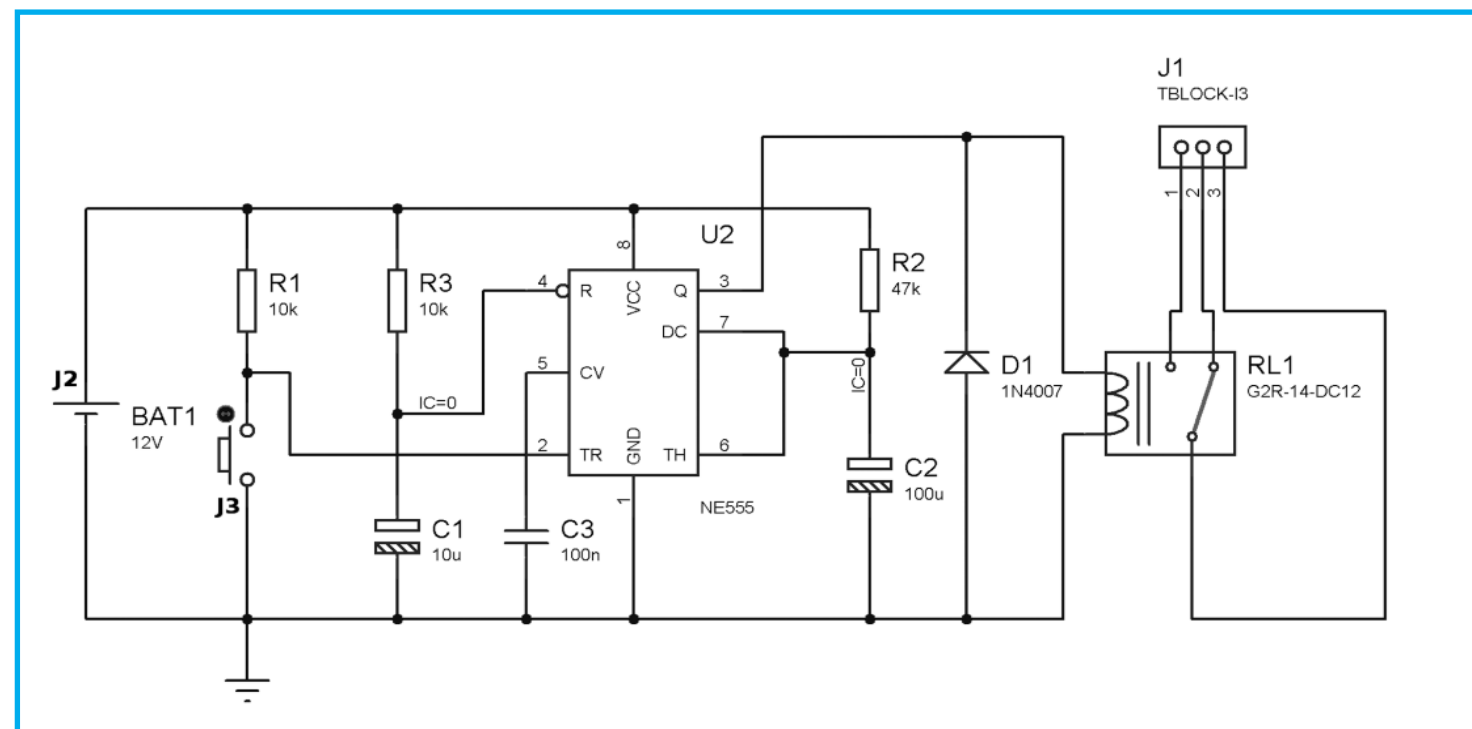



Figura 3: Schema elettrico

Per allungare i tempi di accensione si può aumentare sia il valore di C2 che quello di R2, agendo in questo modo sulla costante di tempo.
Il diodo D1 serve per eliminare le eventuali extratensioni da parte della bobina del relè.

IL PCB

Per il progetto si è pensato di realizzare un circuito molto facile, accessibile a tutti, specialmente ai principianti.
La prima operazione da effettuare è quella di approntare il circuito stampato. Dal momento che esso presenta numerose piste, si consiglia di prepararlo con la tecnica della fotoincisione o quella del ferro da stiro, anche se la sua relativa semplicità permette l'utilizzo dei classici pennarelli indelebili da circuito stampato. Se si decide di utilizzare la fotoincisione, occorre acquistare una basetta presensibilizzata e tagliarla nelle dimensioni necessarie. Si deve stampare quindi su lucido trasparente l'immagine delle piste di rame, cercando di "caricare" quanto più inchiostro possibile. La stampa contiene una testo che ha lo scopo di facilitare il suo l'orientamento sulla piastra. Occorre posizionare il foglio trasparente sulla basetta in modo che tale scritta sia leggibile normalmente. Si impressioni quindi la basetta sotto i raggi ultravioletti per qualche minuto e si sviluppi infine in bagno di acqua e soda caustica. Alla fine si rimuova il rame in eccesso immergendo la basetta in un bagno di percloruro ferrico.
Una volta che il circuito stampato è pronto, occorre forarlo con un piccolo trapano a velocità molto elevata (>8000 g/m) con una punta da 0,8mm per le piazzole dei componenti e con una punta da 1mm per i morsetti da c.s.
Le basette presensibilizzate sono spesso costituite di vetronite, la quale consuma le punte del trapano molto velocemente.

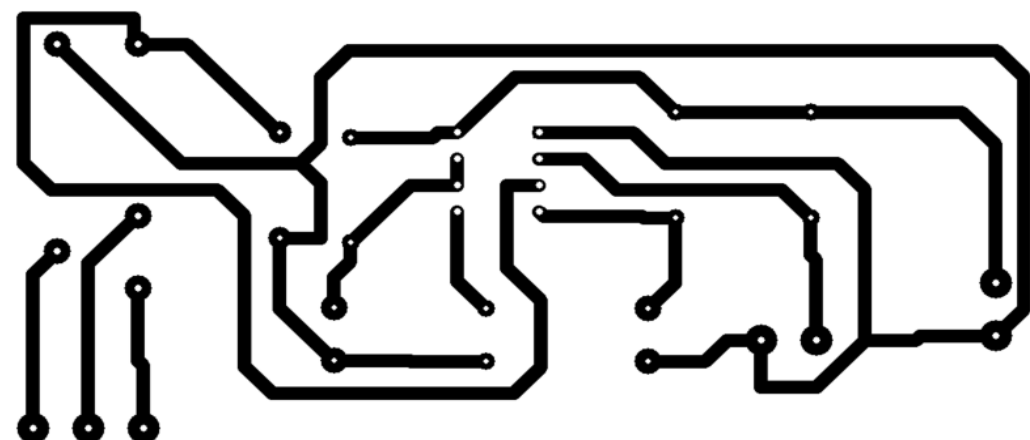



Figura 4: PCB lato rame

MONTAGGIO

Occorre adesso saldare i componenti elettronici sul circuito stampato. Con un saldatore di piccola potenza (circa 30W) si inizi a piazzare quelli a basso profilo, come le resistenze, il diodo e lo zoccolo per l'NE555.

Si passi poi ai componenti più ingombranti, come i morsetti, i condensatori elettrolitici ed il relè.

Occorre saldare con molta precisione e velocità, evitando di surriscaldare gli stessi che mal sopportano le alte temperature. Si verifichi con attenzione anche l'orientamento dei componenti polarizzati.

CABLAGGIO ED UTILIZZO

E' il momento di cablare i componenti e collegarli tra loro. Il morsetto J2 ospita l'alimentazione continua di 12V. Il morsetto J3

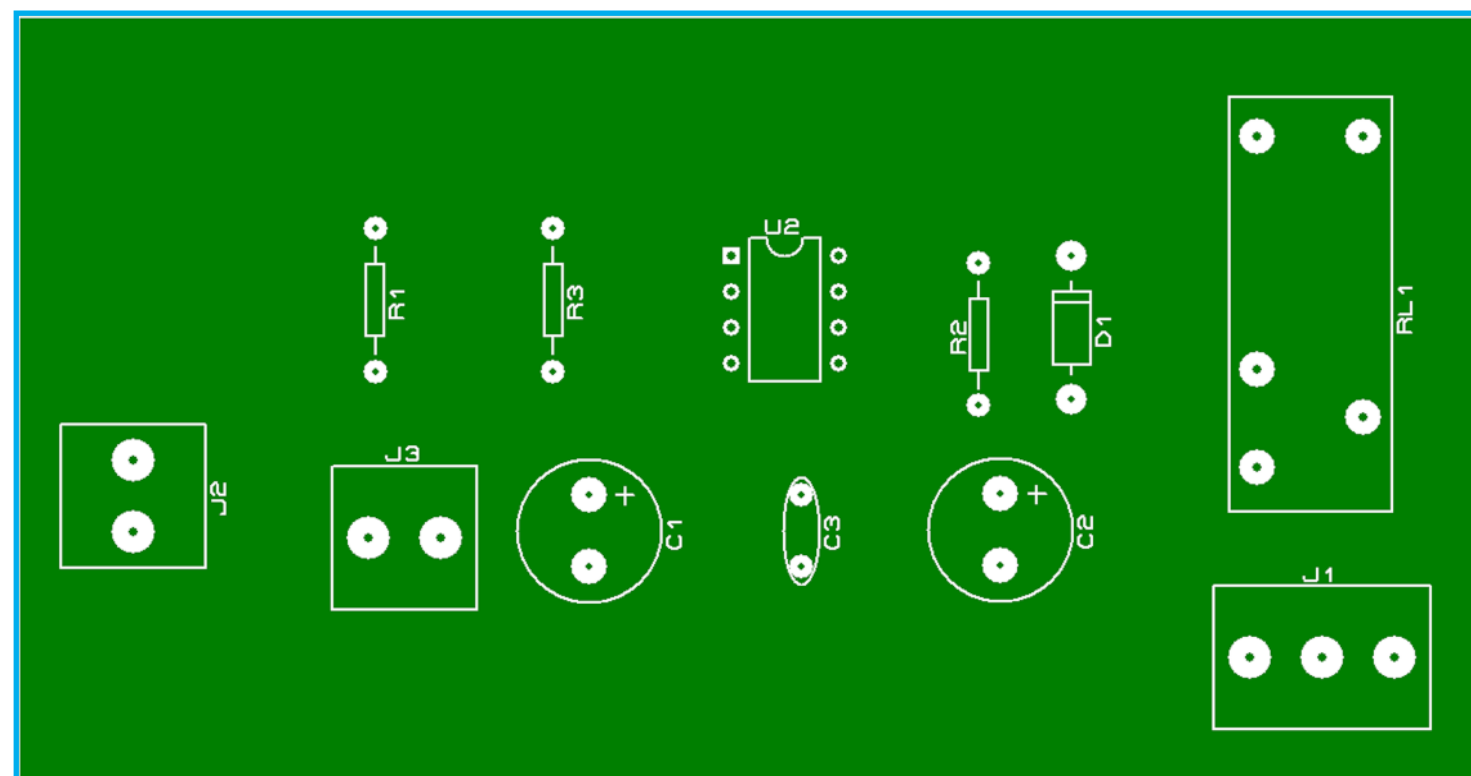



Figura 5: Disposizione dei componenti

Elenco componenti	
R1,R3	10 K 1/4 W
R2	47 K 1/4 W
C1	10 µF 25 V elettrolitico
C2	100 µF 25 V elettrolitico
C3	100 nF
U2	Integrato NE 555 + zoccolo
D1	diodo 1N4007
J1	Morsetto da c.s. - 3 posizioni
J2, J3	Morsetto da c.s. - 2 posizioni
RL1	Relè 12V, 1 scambio 2 posizioni
Piastra di rame	
Alimentazione 12V	
Pulsante normalmente aperto	

è collegato ad un pulsante n/a, che ha la funzione di attivazione temporanea del carico, come spiegato prima. Il morsetto ha tre connessioni. Ne sono utilizzate solamente due, quella comune e quella relativa al contatto normalmente aperto del relè. Prestate sempre la massima attenzione a questa fase, poiché i collegamenti fanno capo direttamente alla 230V domestica. La spina elettrica alimenta la resistenza (o la lampadina o qualsiasi altro carico a 230V), attraverso il relè.

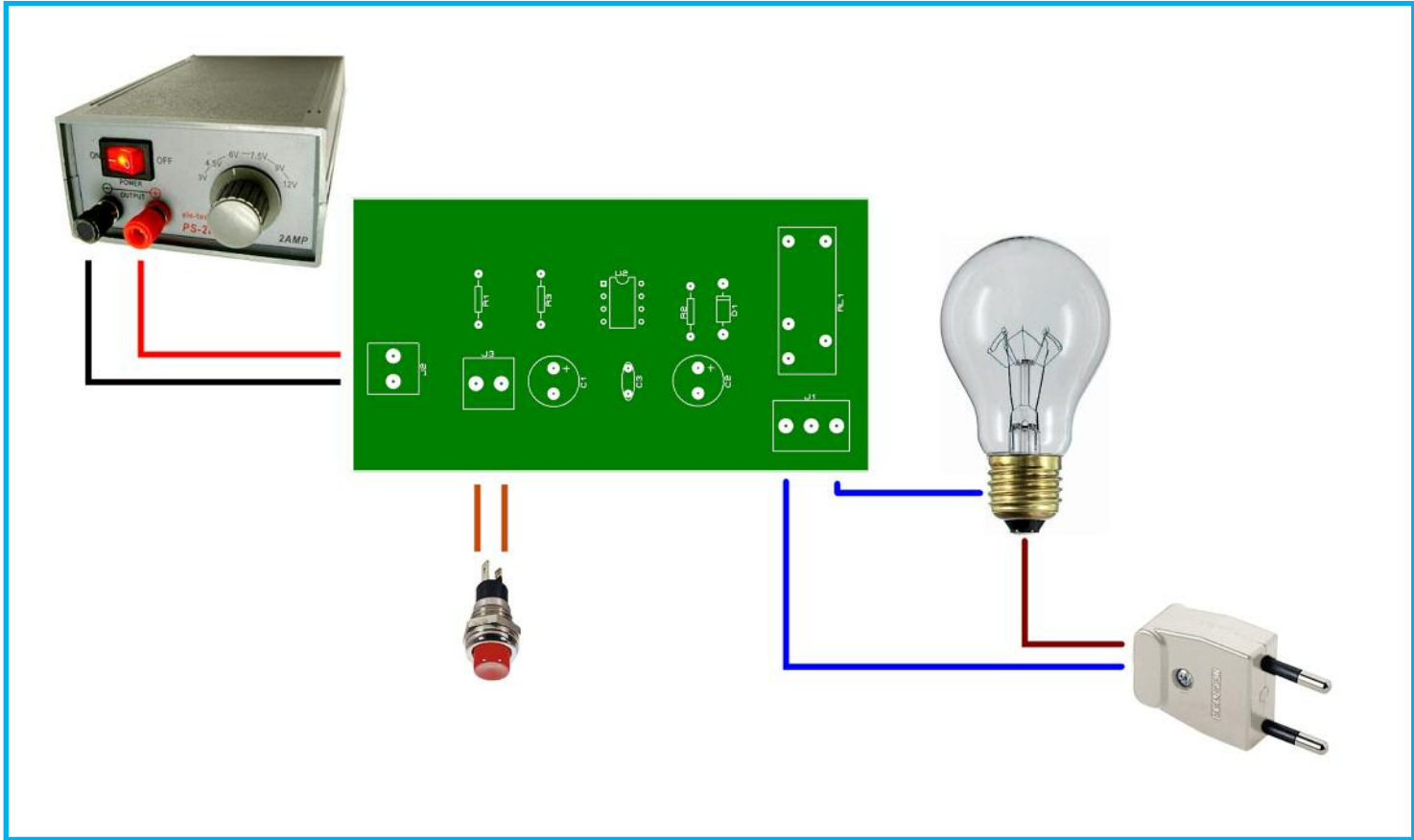



Figura 6: Il cablaggio del sistema

L'INTEGRATO NE555

Il circuito NE555 è un integrato economico, facile da usare, stabile, affidabile e sicuro nei suoi utilizzi. Le applicazioni più importanti sono la generazione di tempi di ritardo, la generazione di oscillazioni ed altro. In pratica, si tratta di un timer di precisione, regolabile per temporizzazioni dal microsecondo (us) fino ad un'ora. Il chip può essere alimentato con tensioni comprese fra i 5 e 15 Vcc ed è in grado di fornire, direttamente all'uscita, una corrente di ben 200 mA.

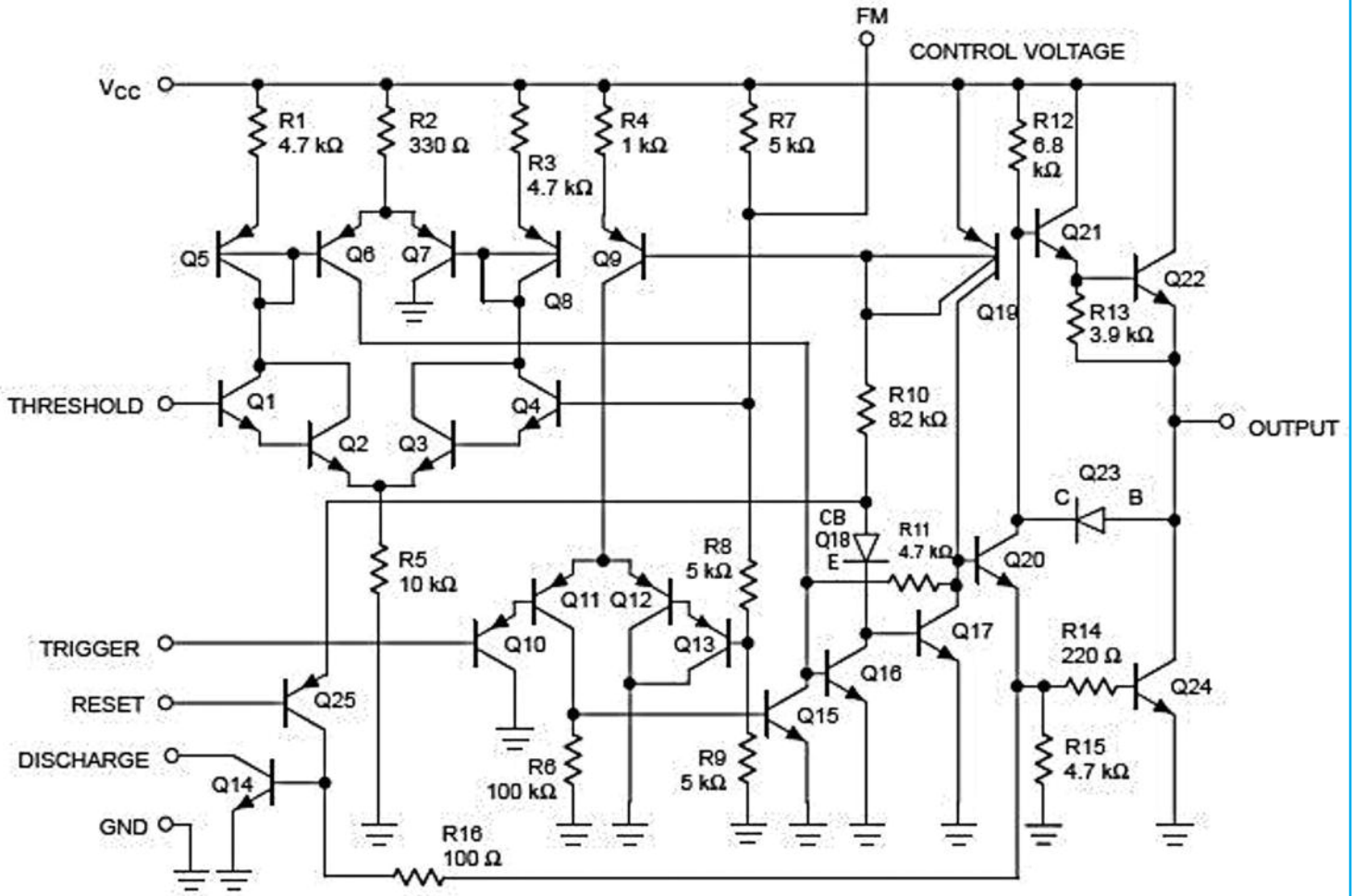



Figura 9: Schema interno dell'integrato NE555

R2	C2	Tempo attivazione
47 k	100 uF	5 secondi
47 k	220 uF	12 secondi
47 k	470 uF	24 secondi
47 k	1000 uF	50 secondi
220 k	100 uF	24 secondi
220 k	220 uF	53 secondi
220 k	470 uF	114 secondi
220 k	1000 uF	240 secondi

Tabella 1: Durata dello stato alto d'uscita, con alcuni valori di R2 e C2

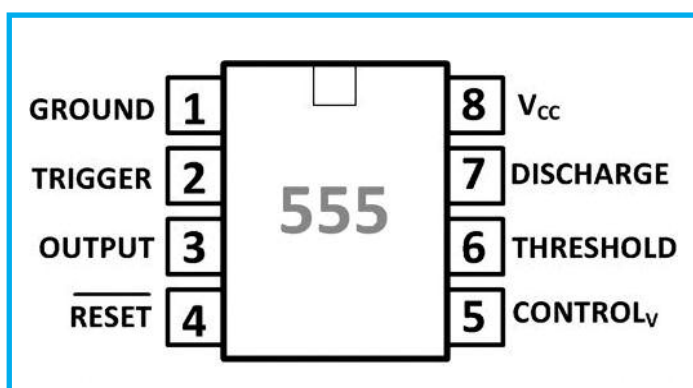



Figura 7: Pinout del 555

L'utilizzo è semplice ed immediato. Effettuati tutti i collegamenti, il sistema è pronto. Inizialmente il carico non è alimentato. Se si preme il pulsante, per un tempo programmato da C2 e R2, il relè viene attivato ed il carico è alimentato. Allo scadere del periodo programmato, la situazione ritorna come quella iniziale, in attesa di un nuovo comando da parte dell'operatore.

TEMPORIZZAZIONE

La durata dello stato di "quasi stabile" dipende dai valori di R2 e C2. Quando, dopo l'attivazione del trigger, il condensatore C2 comincia a caricarsi, l'uscita del 555 si ritrova ad un potenziale alto, sino a quando il potenziale su C2 raggiunge i 2/3 dell'alimentazione elettrica.

La tabella 1 elenca alcuni valori di temporizzazione, in relazione alle diverse combinazioni di R2 e C2.

Esiste comunque una comoda formula per calcolare immediatamente il valore del tempo (espresso in secondi) dello stato di "quasi stabile", fornendo i valori di R2, espresso in k , e C2, espresso in uF:

$$Ts = \frac{R \times C \times 1.1}{1000}$$

Figura 8: Il relè ad uno scambio e due posizioni

Ad esempio, prevedendo una resistenza di 150 k per R2 e un condensatore elettrolitico di 150 uF, il tempo di attivazione dell'uscita del 555 sarà pari a circa 25 secondi. Naturalmente il tempo è riferito al valore "reale" del condensatore e non a quello di targa. Lo stesso dicasi per le resistenze. I primi possono avere tolleranze anche del 30%.

CONCLUSIONI

La realizzazione proposta in queste pagine è semplice ma allo stesso tempo estremamente didattica. Con essa possono essere previste molteplici applicazioni a tempo, come ad esempio quella dell'illuminazione di un locale per pochi secondi o dell'attivazione di una sirena per alcuni istanti.

Occorre prestare molta attenzione alla corretta piedinatura dei relè, considerando anche che i contatti possono essere normalmente aperti e normalmente chiusi. Buon lavoro.

di GRAZIA ANCONA

LEVITATORE MAGNETICO

Un'applicazione elettronica che sfida le più basilari leggi fisiche, ovvero un'occasione per approfondire l'argomento dei controlli automatici e, perché no, divertirci un pochino...

La levitazione di un oggetto, senza ricorrere al "più leggero dell'aria" o a propulsioni varie, è stata veramente risolta solo con l'utilizzo di campi elettromagnetici a forte intensità. Così, fino alla prima guerra mondiale, l'argomento era appannaggio esclusivo di maghi, ciarlatani e, a dirla tutta, di Santi. Ma che cosa è la levitazione magnetica e a cosa può servire? Generando un campo magnetico di opportuna direzione e forza, è possibile contrastare gli effetti della forza di gravità su un oggetto avente proprietà ferromagnetiche. Quest'ultimo rimarrà fermo in un punto d'equilibrio in cui le forze contrapposte, magnetismo e gravità, si annullano reciprocamente. E' importante, come dicevo, che la costituzione dell'oggetto in levitazione abbia proprietà prevalentemente ferromagnetiche, ovvero sia composto (anche in lega) da metalli tipo ferro, nichel o cobalto ecc. L'utilizzo più eclatante di tale tecnica è la riduzione dell'attrito volvente. Chi non conosce il famoso treno a levitazione magnetica su monorotaia? Altre applicazioni riguardano gli assi di trasmissione dei generatori eolici e ammortizzatori per piccoli carichi. Un altro interessante utilizzo della levitazione magnetica è la fusione in aria di materiali che, dato l'elevato grado di purezza richiesto, non possono toccare alcun recipiente. L'applicazione che presenterò non è soltanto un interes-

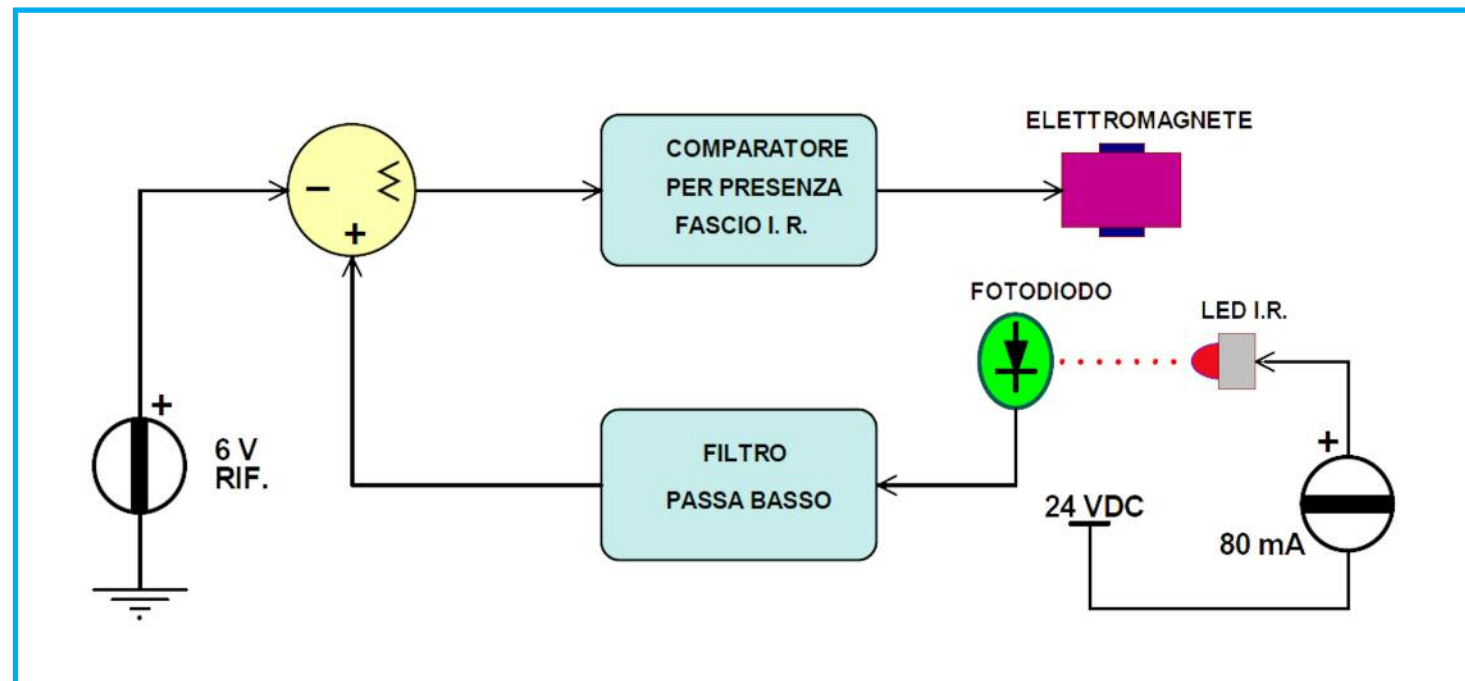



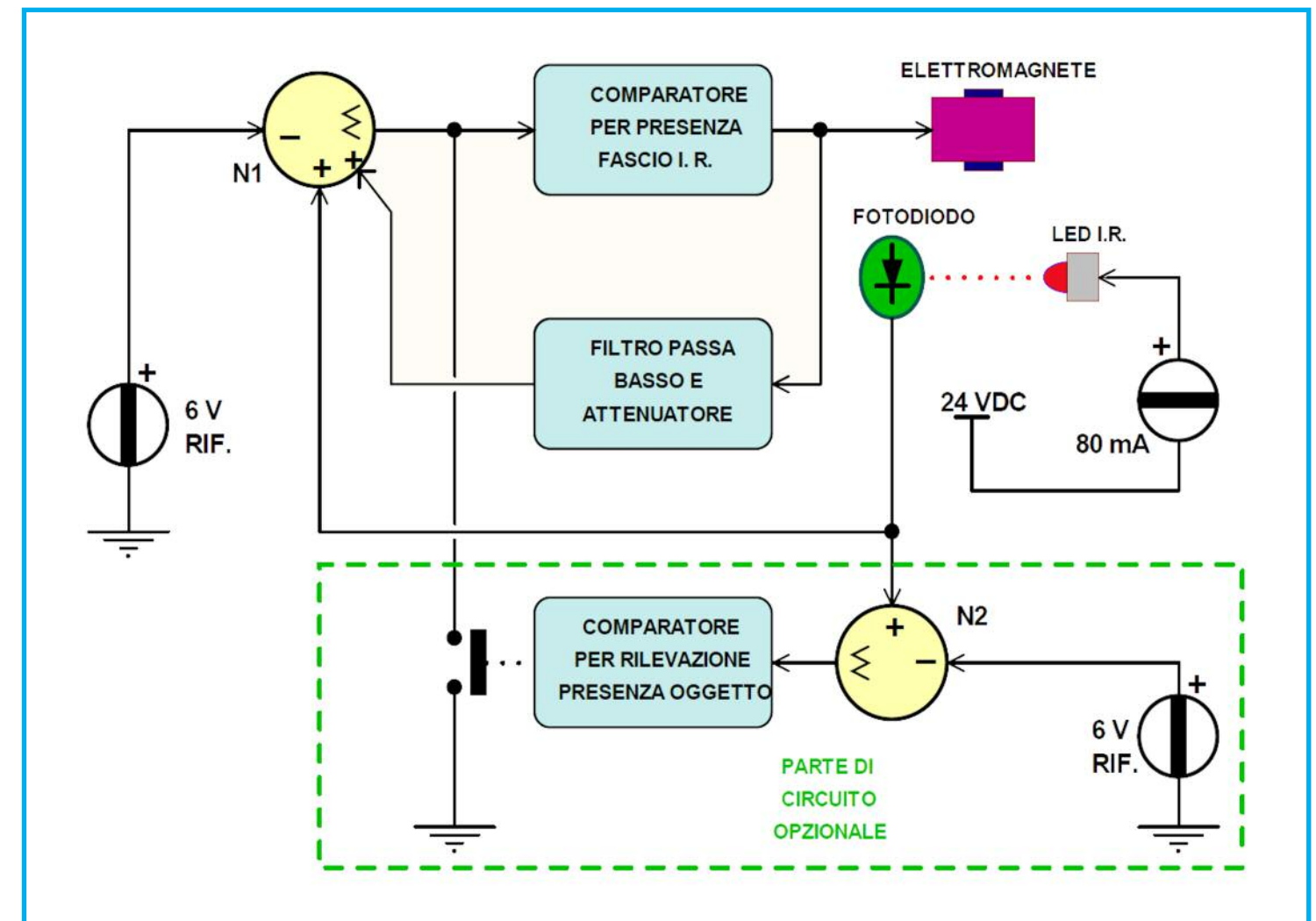
Figura 1: Schema a blocchi prima versione

sante giocattolo ma, contestualmente, una piccola porticina aperta per chi vuole approfondire e studiare questa tecnologia. In aggiunta, dal punto di vista puramente elettronico, l'oggetto presentato rappresenta un interessante modello di studio per chi vuole cimentarsi nel sempreverde settore dei controlli automatici.

PRINCIPIO DI FUNZIONAMENTO

Il primo approccio fu abbastanza intuitivo. In definitiva il problema era semplice: occorreva comandare l'elettromagnete soltanto quando l'oggetto usciva dal fascio infrarosso, per effetto della forza di gravità. A questo punto un comparatore, opportunamente pilotato da una tensione di riferimento, rilevava il cambiamento per mezzo di un fotodiodo e portava in conduzione un elemento di potenza (MOSFET canale N) in grado di fornire tensione (e quindi corrente) all'elettromagnete. Quest'ultimo attirava a sé l'oggetto, "fermando" in tal modo gli effetti dell'attrazione gravitazionale e, nel contempo, faceva sì che l'oggetto interrompesse nuovamente il fascio provocando, per mezzo del comparatore sopraddetto, il rilascio dell'energia magnetica. Questo ciclo avrebbe dovuto ripetersi a velocità sostenuta con l'oggetto che, pur vibrando leggermente, sarebbe rimasto a "galleggiare" nell'aria. Insomma, nella mia testa vi era lo schema a blocchi di figura 1. I più smaliziati avranno senz'altro riconosciuto un banale loop di regolazione automatica (raffigurato in forma canonica) il cui blocco di condizionamento del segnale di controreazione era un semplice filtro passa basso. Ma le cose andarono diversamente. L'oggetto "levitante" non riusciva a stabilizzarsi e finiva per essere attirato dall'elettromagnete oppure cadeva rovinosamente! Forse occorreva essere un po' meno intuitivi e magari un pizzico in più "ragionanti"? Ebbene sì. Probabilmente l'errore era non considerare che l'oggetto, nell'essere attratto verso l'alto, subiva un'accelerazione, seppure di segno contrario alla forza di gravità. Questo fattore non poteva essere controllato dal loop di regolazione schematizzato in figura 1 e finiva per diventare un elemento di instabilità. La soluzione, decisamente meno intuitiva, era quella di trasformare il tutto in una specie di oscillatore fornendo una controreazione di segno positivo. In questo modo, sfruttando l'amplificazione del MOSFET Q1, si realizza una sorta di modulazione a larghezza d'impulso, meglio conosciuta con la sigla PWM. In ragione di ciò, l'oggetto finisce per prelevare dall'elettrocalamita un'energia direttamente proporzionale al suo peso. L'effetto PWM non è del tutto voluto ed accade in virtù del fatto che il MOSFET Q1, con un carico prevalentemente induttivo qual è il nostro elettromagnete, mostra spiccate proprietà di amplificazione in tensione. In altre parole, anche con segnali molto piccoli sul GATE viene raggiunta la saturazione per cui, sul DRAIN, è possibile osservare un'onda quadra. Quest'ultima, in ragione del peso dell'oggetto attratto, varierà il suo ciclo di conduzione. Siamo dunque arrivati allo schema a blocchi definitivo, esso è visibile in figura 2.

Nella versione definitiva, la controreazione positiva giunge al nodo sommatore N1 per mezzo di un filtro passa basso e di un attenuatore resistivo. Essa va a sommarsi (poiché possiede lo stesso segno algebrico) al segnale di controreazione proveniente dal

Figura 2: Schema a blocchi definitivo

* Collegare il capo caldo di C8 al dissipatore di Q1 garantendo il contattto elettrico

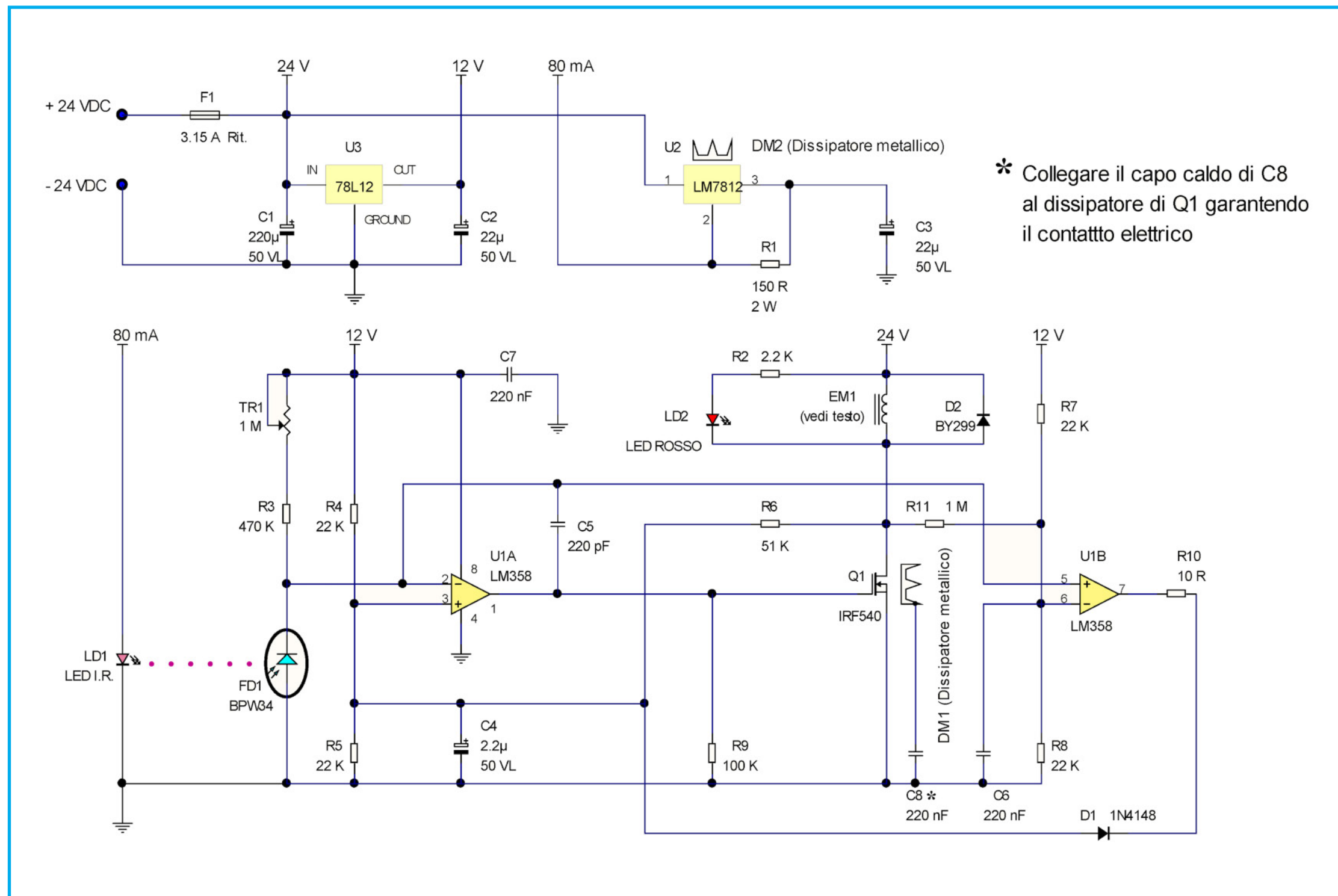



Figura 3: Schema circuitale

fotodiodo. Tale configurazione, per le ragioni spiegate prima, provoca un fenomeno visibilissimo per mezzo di un amperometro posto in serie all'alimentazione. Infatti, variando il peso degli oggetti ferromagnetici levitanti si osserverà un maggiore o minore assorbimento, questo dipenderà dalla massa degli oggetti stessi. I lettori dotati di oscilloscopio potranno invece verificare il fenomeno, in modo alquanto esaustivo, piazzando la sonda direttamente sul DRAIN di Q1. In questa modalità di funzionamento il circuito possiede una stabilità più che sufficiente che però si paga in termini di semplicità, ovvero lo schema a blocchi è un pochino più complesso. La parte tratteggiata in verde merita un discorso a parte. Essa non sarebbe strettamente necessaria poiché il suo unico scopo sta nell'evitare assorbimenti eccessivi di corrente quando all'interno del fascio non vi sono oggetti in sospensione. Infatti, in questa circostanza, l'elettromagnete finirebbe per avere un tempo di conduzione massimo poiché il loop di regolazione tenterebbe di attrarre, con la sua massima energia, un oggetto inesistente. Del resto, quando si sperimenta con numerosi manufatti metallici, magari sotto lo sguardo meravigliato di amici e congiunti, è facile che il fascio resti scoperto anche per interi minuti. Ciò causerebbe il surriscaldamento dell'elettromagnete, del MOSFET Q1 e dell'eventuale alimentatore (se calibrato al minimo). Inoltre, in un momento in cui tutti siamo così attenti allo spreco di energia, mi è sembrato abbastanza opportuno aggiungere questa parte circuitale che, lo ripeto, sarebbe opzionale.

DESCRIZIONE SCHEMA ELETTRICO

Lo schema elettrico, visibile in figura 3, rispecchia quanto descritto nello schema a blocchi. In realtà, per esigenze esemplificative, la fase di alcuni segnali (ovvero i loro segni algebrici) è stata volutamente cambiata e quella reale è, ovviamente, rilevabile dallo schema elettrico.
Nell'elaborare il circuito ho provato, e spero di esserci riuscita, ad utilizzare componenti facilmente reperibili e a ridurre il processo di messa a punto. I comparatori sono i due OP AMP presenti nel popolare LM358, mentre per il fotodiodo si è scelto un comune BPW34, fotodiodo sensibile ai raggi infrarossi e (ahimé) anche ad una certa parte dello spettro visibile. Il diodo emettitore è stato ricavato da un telecomando fuori uso ed è pilotato a corrente costante per una migliore stabilità. Il MOSFET è un comune IRF540 che però va montato su un dissipatore di almeno 50 cmq. Nel montaggio del prototipo (vedi figura 4), anche il dissipatore è stato "arrangiato" da un'altra applicazione che faceva uso di contenitori TO3. I numerosi buchi testimoniano quanto affermo.

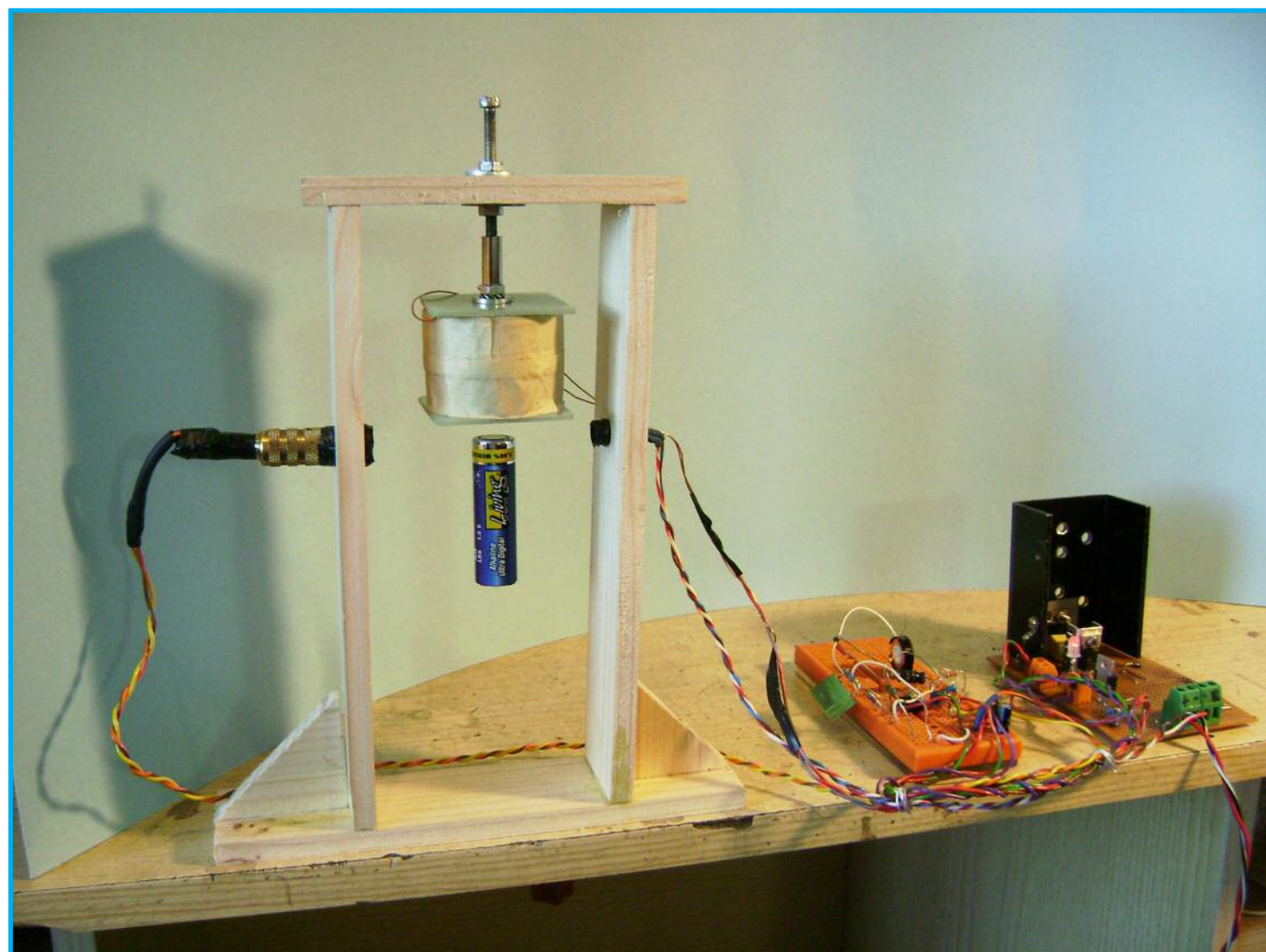



Figura 4: Vista di assieme del prototipo durante il funzionamento

Come si evince dalla foto del montaggio completo, la parte di potenza è stata costruita su bachelite "millefori" mentre la parte di controllo ha trovato migliore sistemazione su una basetta per montaggi sperimentali. Insisto sulle dimensioni del dissipatore di Q1 poiché il MOSFET, quantunque non lavori in zona lineare, tende a scaldare per effetto delle perdite di commutazione a frequenza elevata ed anche per la sua resistenza in conduzione (RDon) che, dato il modello molto economico, non è delle migliori. E' opportuno non connettere elettricamente il dissipatore di Q1 al suo DRAIN. L'aletta dissipatrice di quest'ultimo andrà isolata (un foglio di mica con rondella isolante plastica è sufficiente) per evitare che il tutto diventi un'efficace antenna capace d'irradiare la frequenza di commutazione e quindi di essere fonte inequivocabile d'instabilità. Per migliorare l'effetto schermante è anche necessario porre a massa (per i segnali) il dissipatore stesso. Di ciò si incarica il condensatore C8. Anche il regolatore di corrente U2 deve avere un piccolo dissipatore, all'uopo è sufficiente una piccola aletta. Il condensatore C4 realizza, insieme al partitore resistivo R4 - R5 e alla resistenza R6, il filtro passa basso che produce la reazione positiva. Insieme all'elettromagnete, è il componente più critico di tutto il sistema. Tra l'altro, i due citati oggetti possono, in una certa misura, essere interdipendenti. Questo significa che eventuali tolleranze costruttive di EM1 possono far variare il valore di C4. In pratica, durante il collaudo, se si riscontra un'instabilità eccessiva, è possibile variare il valore di C4 portandolo sperimentalmente da 1 uF a 4,7 uF. Dicevamo della criticità dell'elettromagnete EM1. Per la verità, durante le prove, ho sperimentato avvolgimenti di campo "lobotomizzati" da relè in disuso. Per utilizzarli ho semplicemente asportato il gruppo di contatti e la struttura metallica portante. Si trattava di relè a 6 VDC aventi portata pari a 15 A sui contatti, ovvero componenti molto robusti destinati al controllo di lampade, motori in CA, ecc. Sebbene funzionassero, pur accettando un surriscaldamento della bobina di campo al limite della rottura, non mi hanno soddisfatto del tutto. Ho quindi cercato di avvolgere un "mio" elettromagnete i cui dati costruttivi sono riportati in figura 5.

Pur non avendo alcuna esperienza di avvolgimenti (!), il risultato è stato buono e degno di essere imitato da tutti coloro che non hanno paura di mettere una spira dietro l'altra per... 1300 volte! L'elettromagnete così costruito ha dato risultati di robustezza, trazione e stabilità eccellenti, con filo da 0,55 mm di diametro (ma va bene anche da 0,6 o da 0,62) la sua resistenza ohmica è pari a circa 10,5 Ohm e questo, lo ripeto, per un totale di 1300 spire. In realtà, secondo le tabelle di progetto, il filo da 0,55 non sarebbe del tutto idoneo poiché ha una corrente massima ammissibile leggermente inferiore a quella necessaria per la levitazione di oggetti con massa su-

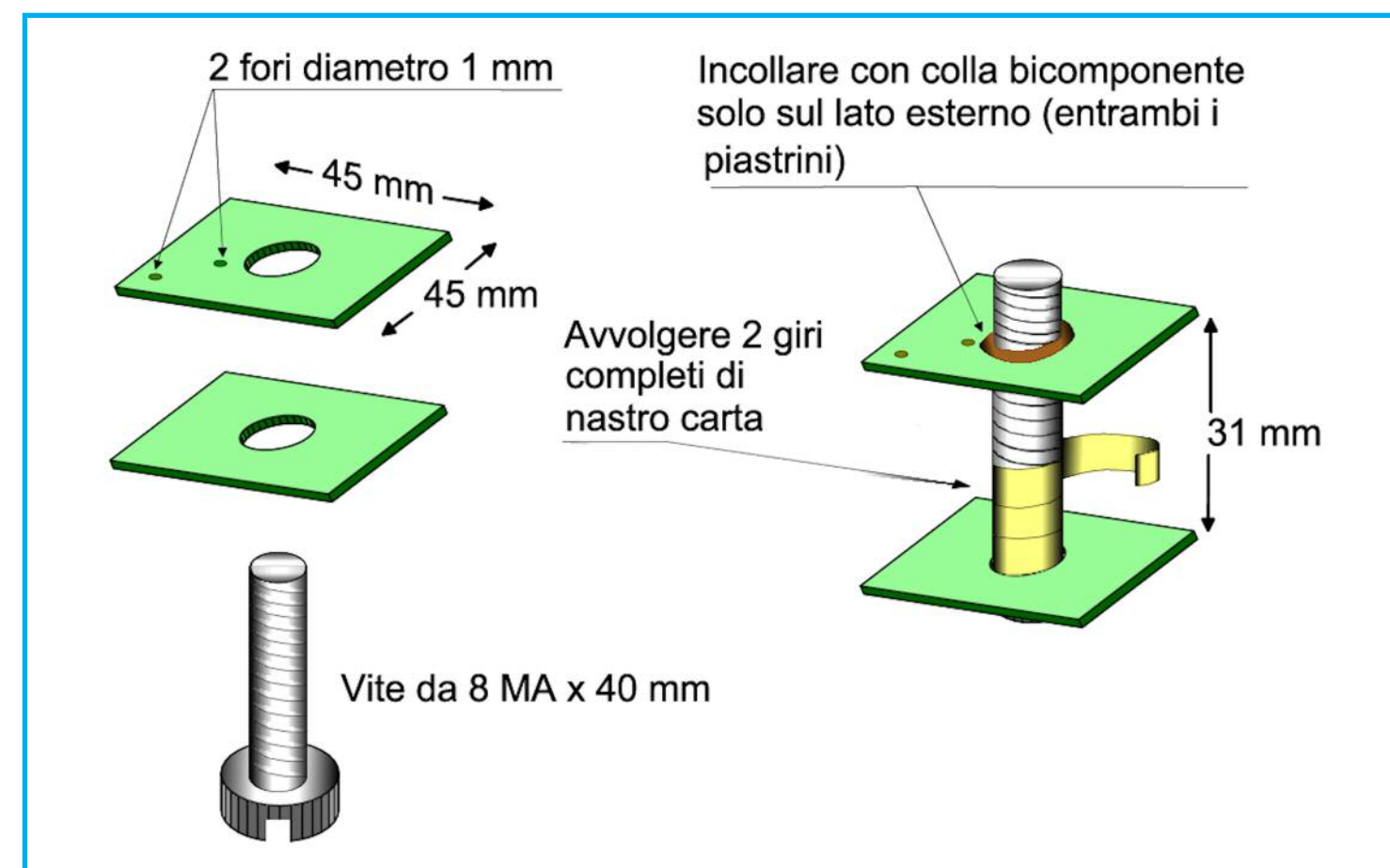



Figura 5: Particolari costruttivi dell'elettromagnete

periore a 50 grammi. Tuttavia, durante le prove, si è constatato che il problema non è poi così evidente per un funzionamento discontinuo. Ogni strato di avvolgimento è stato fermato con una piccola strisciolina di nastro-carta (quello che usano i … carrozzieri!) avente altezza pari all'avvolgimento stesso (31 mm). Non ho ritenuto opportuno isolare completamente ogni strato perché il nastro carta ha uno spessore non indifferente ed, alla fine, sarebbero entrate molte meno spire. Se al posto del nastro-carta si fosse usato un apposito materiale, come il KAPTON ad esempio, sarebbe stato certamente possibile isolare ogni strato con enormi benefici ai fini del riscaldamento totale. Quanto esposto è solo un esempio, i lettori più esperti potranno escogitare soluzioni diverse da quelle indicate in figura 5, forse anche più semplici ed efficaci. Le due guanciole quadrate costituenti il "pacco" devono essere ritagliate da un materiale termoindurente (io ho usato la vetronite) poiché tutto l'avvolgimento, nucleo compreso, tende a scaldare. Se si usa la vetronite ramata (su un solo lato) è necessario che la parte in rame sia, ovviamente, all'esterno dell'avvolgimento. Il nucleo, una comune vite in ferro a testa cilindrica da 35 x 8MA, andrà incollata alle guanciole sulla parte esterna. Durante il tempo d'incollaggio, che sarà fatto con adatta colla bi-componente, le guanciole stesse saranno fermate sulla vite con qualche giro di nastro che sarà poi asportato. Parliamo ora del "front-end" ottico. Il fotodiodo FD1 è polarizzato in senso "inverso" dal trimmer TR1 e dal resistore R3, il trimmer sarà l'unico componente che andrà regolato, come vedremo, in fase di messa a punto. In questa configurazione, il fotodiodo funziona nella regione dove prevalgono le cariche minoritarie quindi la debole corrente che vi scorre aumenta se esso è colpito dal fascio infrarosso. Il segnale in tensione risultante, ad alta impedenza, pilota 2 ingressi sui comparatori che, avendo essi stessi impedenza altissima, non disturbano troppo il funzionamento del fotodiodo. Il resistore R11 provvede a fornire un minimo di isteresi per il comparatore destinato a scoprire la mancanza dell'oggetto levitante (ovvero U1B). Infatti, quando il dispositivo funziona a vuoto, la tensione sul pin 6 (ingresso invertente) fornita dal partitore R7 – R8 è decisamente più bassa di quella fornita dal fotodiodo. Il comparatore porterà quindi la sua uscita a circa 0 V (saturazione negativa). In queste condizioni D1 sarà polarizzato direttamente ed andrà in conduzione portando il pin 3 del primo OP AMP (ingresso non-invertente) ad una tensione prossima allo zero. In virtù di ciò anche l'uscita di U1B andrà in saturazione negativa interdicendo il MOSFET ed annullando in tal modo la corrente nell'elettromagnete. Non appena il fascio si interrompe la tensione sul pin 3 di U1A tende a prevalere e così sarà anche per la tensione sul pin 5 di U1B, in tal modo l'uscita di quest'ultimo andrà in saturazione positiva (circa 10,5 VDC). Tuttavia, essendo ora il diodo D1 polarizzato inversamente, il nodo di controreazione (N1 nello schema a blocchi) non sarà affatto perturbato ed il loop principale potrà funzionare. E' opportuno, nel montaggio, che i condensatori C5 e C7 siano posti il più vicino possibile ad U1A-B, questo discorso è quasi inopinabile per

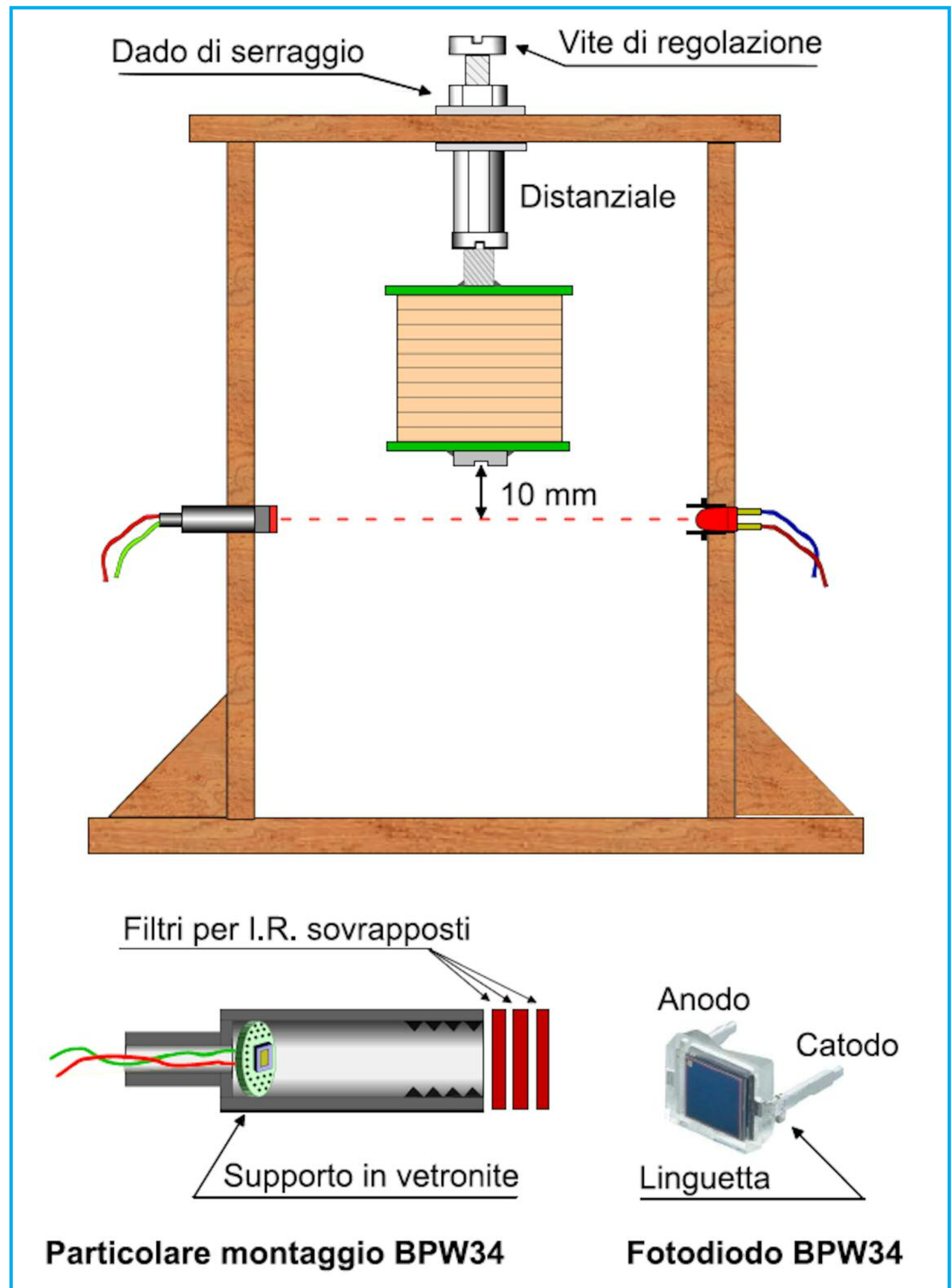



Figura 6: Struttura del gruppo elettromagnete, fotodiodo e LED I.R.

C5 che serve a scongiurare eventuali oscillazioni in alta frequenza. L'alimentazione generale del dispositivo deve essere fatta con una sorgente che abbia una stabilizzazione almeno del 5% con una corrente minima pari a 1,6 A. Questo significa che la variazione ammessa sulla tensione nominale deve essere contenuta fra 25,2 V e 22,8 V. Non ho ritenuto necessario presentare anche la sezione alimentatrice: veda il lettore di risolvere come meglio crede, anche con l'autocostruzione se necessario. Tuttavia, molti alimentatori switching economici (di tipo plug-in) soddisfano le esigenze richieste per questa applicazione oppure, come nel mio caso, si ricorrerà al classico alimentatore da laboratorio. Se non sarà montata la parte di circuito opzionale, la corrente, con il dispositivo funzionante a vuoto, potrà raggiungere anche 2,5 A. Attenzione quindi perché la potenza totale fornita, in quest'ultimo caso, sarà pari a ben 60 W. Il led LD1 ha il solo scopo di fornire un monitoraggio visivo di ciò che sta accadendo. Durante il funzionamento non dovrà mai accendersi a luce piena mentre, senza oggetti in sospensione, deve essere quasi completamente spento (è ammessa una debolissima emissione dovuta a perdite resistive del MOSFET). In fase di attrazione, l'elettromagnete può emettere un leggero fischio e la cosa è del tutto naturale. Pazienza, non è possibile ottenere da una costruzione artigianale un elemento del tutto silenzioso!

MONTAGGIO DEL DISPOSITIVO

Il gruppo elettromagnete, diodo emettitore e fotodiodo devono essere montati sullo stesso supporto con gli ultimi 2 elementi in allineamento che, per quanto possibile, deve essere abbastanza accurato. Come accennato, il fotodiodo è leggermente sensibile anche alla luce ambiente, soprattutto se quest'ultima è prodotta da lampade ad incandescenza o da raggi solari diretti. Ciò implica una schermatura che, nel prototipo, è stata ottenuta con 3 fogli di acetato sovrapposti tolti da un display a LED rossi ormai fuori uso. L'acetato in oggetto serviva appunto da filtro per frequenze ottiche prossime all'infrarosso, tuttavia non escludo che i piccoli schermi di PERSPEX o di policarbonato, di solito usati nei telecomandi ad I.R. modulati, possano far di più e meglio. Per aumentare l'insensibilità alla luce ambiente il fotodiodo è stato poi posto all'interno di un JACK in modo che solo il fascio del diodo emettitore possa colpire la sua superficie. Anche la sistemazione all'interno del JACK è stata dettata da motivi pratici (era disponibile nel solito cassettone..) per cui, se il lettore ha di meglio, può benissimo tentare altre soluzioni. Per semplificare il posizionamento del fotodiodo è a mio parere necessario, qualunque sia il supporto scelto, montarlo su un piccolo ritaglio rotondo di vetronite. Questa soluzione impedirà la rottura dei sottilissimi reofori e renderà il tutto più affidabile nel tempo. Il diodo emettitore è invece fissato ad un classico porta LED dotato di ghiera di tenuta a clips (per elementi da 5 mm). La ghiera, in verità, l'ho usata per "restringere" leggermente il fascio in modo da aumentare ancora la stabilità. dell'insieme. Al posto della ghiera di fissaggio possiamo però usare qualunque tondino di materiale diamagnetico, l'importante è che sia perfettamente combaciante con il bordo del LED ed abbia una lunghezza non superiore a 5 mm. Usando il legno di abete come materiale per la costruzione del supporto, possiamo sfruttare l'incastro dei vari elementi nei corrispondenti fori senza preoccuparci troppo della tenuta meccanica. Nonostante le precauzioni sopraviste, è opportuno non esporre a luce ad incandescenza diretta l'ingresso del fotodiodo durante il funzionamento. Tenere quindi le lampade da tavolo ad una certa distanza oppure orientarle altrove! L'elettromagnete sarà fissato al centro esatto della distanza fra LED emettitore e fotodiodo. Nel prototipo è stato previsto un sistema di regolazione in modo che sia possibile avvicinarlo più o meno al fascio I.R. Esso è costituito da un distanziale esagonale (ma può essere anche tondo) avente doppia impanatura femmina da 3MA e dovrebbe essere in materiale diamagnetico (alluminio, ottone, ecc.). La cosa non è strettamente necessaria ma, se soddisfatta, diminuisce la lunghezza delle linee di campo aumentando, seppure di poco, la potenza attrattiva. Di solito, la distanza dell'estremità dell'elettromagnete (testa della vite costituente il nucleo) dal fascio I.R. è compresa fra 8 - 11 mm e potrebbe essere ritoccata in fase di messa a punto. Il supporto, come accennato, è stato fatto ricorrendo al legno di abete ma anche in questo caso il lettore può utilizzare altri materiali, anche alluminio (per l'amor del cielo non la lamiera di ferro!). Il cavo che collega il fotodiodo al circuito potrà anche

Figura 7: Dettaglio di fissaggio dell'elettromagnete

essere schermato (con la calza a massa) mentre il cavo che alimenta l'elettromagnete dovrà essere di sezione adeguata (almeno 0,8 mmq). In figura 6 è possibile vedere il disegno della struttura del complesso mentre, in figura 7, possiamo vedere il dettagli di montaggio dell'elettromagnete.

TARATURA E RISOLUZIONE INCONVENIENTI

Come accennato, ho cercato di risolvere le operazioni di taratura con la regolazione di un unico trimmer: TR1. Questi regola il punto di lavoro del fotodiodo in modo che l'eventuale variazione di potenziale ai capi di quest'ultimo sia sufficiente a far cambiare stato al comparatore. Per la regolazione procediamo scollegando da un solo capo il diodo D1 e facciamo lo stesso con uno dei due fili che collegano l'elettromagnete al circuito. Quindi "oscuriamo" con un cartoncino nero il diodo emettitore I.R. oppure scolleghiamo la sua alimentazione. In questo modo il fascio I.R. ed il circuito rivelatore della mancanza di oggetti levitanti non influiranno sulla taratura. Applichiamo ora un tester digitale sul pin 3 di U1. Essendo un punto ad alta impedenza non è possibile usare il solito tester analogico da 20 KOhm/V (ammesso che vi sia rimasto ancora qualcuno a possederlo…). Prendere nota della lettura che dovrebbe aggirarsi, se tutto va bene, intorno ai 6 V. Porre ora il tester sul pin 2 di U1 e tarare TR1 per una tensione che sarà:

Tensione su pin 2 = (Tensione su pin 3) – 0,5 V

Durante la taratura il fotodiodo dovrà, per quanto possibile, operare nelle condizioni di luminosità a cui sarà sottoposto nel normale funzionamento. E' probabile che si renderà necessario qualche piccolo ritocco al trimmer TR1 durante le prime prove. La fase di collaudo dovrebbe prevedere la levitazione di piccoli oggetti con diverso peso, tuttavia procedendo con i primi aggiustamenti di TR1 è importante che l'oggetto non sia né troppo pesante né troppo leggero. Un peso di circa 45 grammi (una batteria tipo "AA", ad esempio) dovrebbe andare bene. Nella fase iniziale delle prove è opportuno che la distanza fra elettrocalamita e fascio non sia superiore ad 8 mm. Se tutto va bene sarà possibile aumentare gradatamente questa distanza, beneficiando, in tal modo, di un effetto levitante più "scenico". Per oggetti molto pesanti o molto leggeri potrà essere necessario un leggero ritocco di TR1 e si tenga comunque presente che oggetti superiori a 70 – 80 g possono essere attratti solo con l'elettrocalamita molto vicina al fascio. Gli inconvenienti più comuni a cui si può incorrere possono essere:

- Scarsa stabilità (oggetti attratti completamente o cadenti)
- Surriscaldamento elettromagnete
- Elevato assorbimento

La stabilità, se tutto è stato fatto a dovere, dipende molto dalla costruzione dell'elettromagnete e dalla capacità C4. Anche un elettromagnete troppo lontano o troppo vicino al fascio può dar luogo a fenomeni d'instabilità, specie a caldo. E' possibile variare il valore di C4 anche mettendo in parallelo altre piccole capacità ed, al limite, provare a variare il valore di R6 di qualche KOhm in più o in meno. Verificare, in ogni caso, se la tensione di alimentazione è abbastanza stabile e con ripple contenuto (max 0,5 Vpp). Per diminuire il ripple è possibile aumentare il valore di C1 ma la misura può essere fatta solo con un oscilloscopio. Anche eventuali auto-oscillazioni di U1 possono provocare instabilità. Se necessario aumentare il valore di C5 (fino a 560 pF) e/o mettere un condensatore ceramico da 220 pF fra il pin 6 e 7 di U1. Ricordare che un cablaggio fatto male (loop di massa, piste lunghe, ecc.) può dar luogo a questi inconvenienti. Il surriscaldamento dell'elettromagnete può accadere soprattutto se usiamo bobine di campo di relè a 6 VDC o 12 VCA. In questi casi c'è poco da fare, o proviamo con un'altra bobina o diminuiamo la tensione di alimentazione anche fino a 18 VDC. L'elevato assorbimento può dipendere anch'esso da bobine di campo non idonee o da auto-oscillazioni in alta frequenza. Per queste ultime si procederà come nel caso d'instabilità.

MIGLIORIE E ULTERIORI SPERIMENTAZIONI (OVVERO L'APPETITO VIEN MANGIANDO…)

Premesso che quanto esporrò non è stato del tutto verificato, vorrei suggerire qualche indirizzo da seguire a chi si appassionerà a questo tipo di applicazione. Durante i vari test, come dicevo, si è evidenziata una certa dipendenza del fotodiodo dalla luminosità ambiente. E questo nonostante la luce fosse filtrata da ben 3 strati di acetato rosso-intenso. E' possibile ridurre il fenomeno, e nel contempo semplificare la costruzione, utilizzando 2 fotodiodi simili. Lo schema di principio mostrato in figura 8.

Il circuito di principio di figura 8 permette la

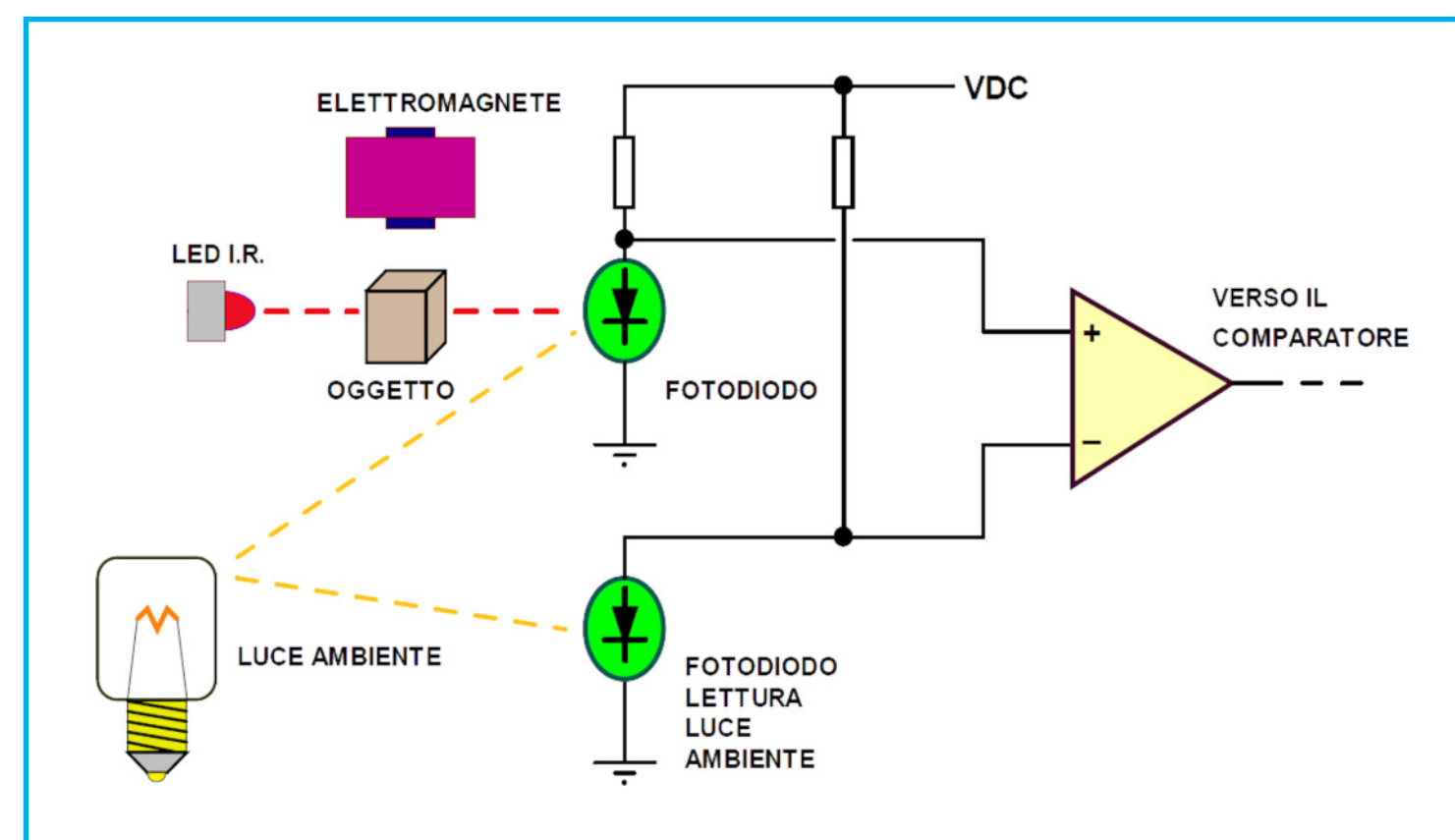



Figura 8: Schema di principio con 2 fotodiodi

"sottrazione" degli effetti perturbativi della luce ambiente al segnale prodotto dal fascio I.R. In tal modo, solo l'effetto del fascio I.R. sarà elaborato ai fini del controllo. Questa tecnica permette di lasciare i fotodiodi privi di qualunque filtro ma, per contro, devono comunque essere posizionati in modo che, senza l'oggetto levitante, entrambi possano essere colpiti dal fascio I.R. Adottando questa soluzione, lo schema circuitale si complicherebbe per la presenza di un ulteriore OP AMP. L'altra miglioria riguarda l'elettromagnete. E' possibile utilizzare per quest'ultimo un vecchio trasformatore da 3 - 5 W a 50 Hz privo di parte dei lamierini (a nucleo aperto). Si procede in questo modo: una volta reperito il trasformatore lo si smonta completamente togliendo il pacco lamellare (attenzione che può essere resinato!) senza danneggiare il cartoccio. I lamierini a forma di segmento (traversini) andranno quindi eliminati mentre saranno conservati quelli a forma di "E" rovesciata. Di solito il pacco lamellare è montato in modo che traversini ed "E", per migliorare la loro tenuta, siano alternati. Questo significa che tutto il pacco dovrà essere smontato e poi, una volta terminato di riavvolgere il cartoccio, opportunamente rimontato (con i soli lamierini a forma di "E"). A questo punto è possibile svolgere completamente i vecchi avvolgimenti presenti sul cartoccio e, con santa pazienza, si prova ad avvolgere almeno 800 spire di filo smaltato da 0,55 – 0,62 mm di diametro (il numero delle spire è indicativo). Una volta terminato, come già detto, si rimonta il pacco lamellare facendo in modo che un'estremità del trasformatore sia ora libera dai traversini rettangolari (vedi figura 9).

Naturalmente la parte priva di nucleo andrà rivolta verso il fascio I.R. Questa modifica è mirata ad ottenere una distanza maggiore dell'oggetto levitante dall'elettromagnete, inoltre dovrebbe diminuire la corrente fornita dall'alimentatore (a parità di peso dell'oggetto). Questi vantaggi sono spiegabili dal minor cammino in aria che dovranno fare le linee di forza magnetica generate. L'ultima modifica è puramente "sensoriale" e serve a far saltellare dolcemente, di qualche millimetro, l'oggetto in levitazione. Posso garantirvi che l'effetto è abbastanza simpatico, dunque vale la pena di provare! Per ottenerlo è sufficiente "modulare" il pin 2 di U1A con un segnale di 0,1 – 0,18 Vpp (senza DC sovrapposta) a frequenza molto bassa (circa 0,5 Hz) in modo che l'oscillazione sia ben visibile. Qualunque funzione (sinusoidale, rettangolare, ecc.) può andare bene. Buon lavoro!

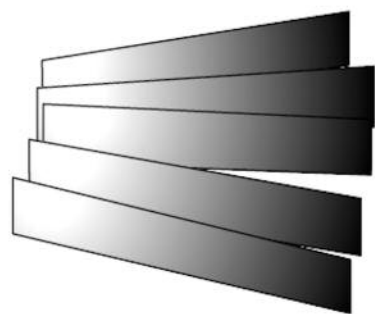
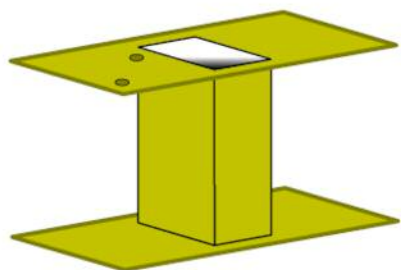
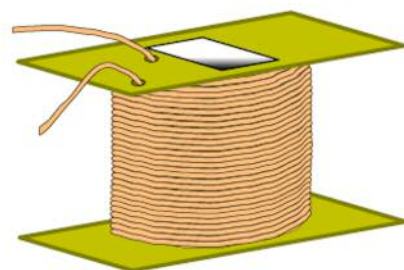
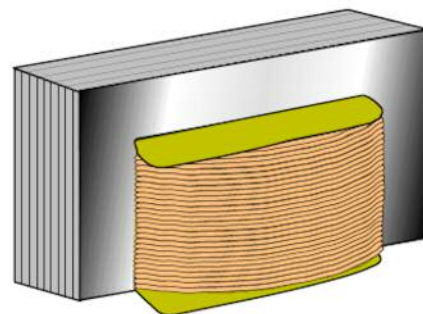



Figura 9: Modifica di un trasformatore per funzionare da elettromagnete

ELENCO COMPONENTI	
C1	Cond. elettrolitico 220 uF 50 VL
C2; C3	Cond. elettrolitico 22 uF 50 VL
C4	Cond. elettrolitico 2,2 uF 50 VL
C5	Cond. ceramico a disco 220 pF 50 VL
C6, C7, C8	Cond. poliestere o multistrato 220 nF 50 VL
D1	Diodo 1N4148 o 1N4149
D2	Diodo BY299 (oppure BY399, BYW56, ecc)
DM1	Dissipatore metallico di almeno 50 cmq per TO220
DM2	Piccolo dissipatore metallico per TO220 (altezza 15 mm circa)
EM1	Elettromagnete (vedi testo)
F1	Fusibile ritardato da 3,15 A in vetro
FD1	Fotodiodo tipo BPW34
LD1	LED da 5 mm per infrarosso (vedi testo)
LD2	LED rosso da 3mm
Q1	MOSFET canale N tipo IRF540 (TO220, montato con dissipatore DM1 previo isolamento del DRAIN mediante mica ed isolatore passante)
R1	Res. 150 Ohm 2 W (può essere anche a filo)
R2	Res. 2,2 KOhm ¼ W
R3	Res. 470 KOhm ¼ W tolleranza 1% a film metallico
R4; R5; R7; R8	Res. 22 KOhm ¼ W a film metallico
R6	Res. 51 KOhm ¼ W tolleranza 1% a film metallico (vanno bene anche 2 resistori posti in serie da 47 K e 3,9 K)
R9	Res. 100 KOhm ¼ W
R10	Res. 10 Ohm ¼ W
R11	Res. 1 MOhm ¼ W
TR1	Trimmer 1 MOhm (evitare di montare modelli sub-miniatura ed usare solo tipi di buona qualità)
U1A-B	OP AMP tipo LM358A (tipo dual in line ad 8 pin)
U2	Regolatore LM7812 (TO220, montato con dissipatore DM2)
U3	Regolatore 78L12 (TO92, va ben anche LM7812 se non si hanno problemi di spazio)

di ANTONIO GIANNICO

CORSO MIKROPASCAL PER PIC
OROLOGIO CON DISPLAY 7 SEGMENTI

(parte decima)

Un orologio digitale basa il suo funzionamento su un clock, la visualizzazione sull'impiego di un display e la programmazione sull'impiego di pulsanti che consentono di modificare l'ora corrente in modo da riallinearla, se necessario, con quella di un orologio di riferimento. Dedicheremo questa lezione all'implementazione, in codice MikroPascal, di questa utile applicazione utilizzando il solito microcontrollore PIC e alcuni display a 7 segmenti

In questa decima lezione del corso non introdurremo argomenti nuovi dal punto di vista teorico. La lezione sarà piuttosto occasione per applicare, in un progetto un po' più impegnativo di quelli mostrati fino a questo momento, molto di quanto appreso nelle precedenti lezioni. In particolare utilizzeremo le nozioni su I/O, gestione di display a sette segmenti e interrupt per implementare un orologio programmabile. L'applicazione potrà essere considerata una base per progetti più ampi da sviluppare successivamente.
Si tratta di una tipica applicazione che necessita di precise temporizzazioni che devono essere mantenute indipendentemente dal flusso di programma mediante un corretto impiego dei timer integrati nel controllore.
L'orologio prevede l'impiego di un display di visualizzazione e pulsanti di programmazione per la correzione dell'orario. L'implementazione firmware è realizzata in codice MikroPascal PRO for PIC intorno al solito microcontrollore PIC16F877 di Microchip.
Alla fine della lezione il lettore disporrà dei sorgenti del firmware e dello schema elettrico dell'applicazione, potrà in questo modo riprodurre l'applicazione stessa in maniera relativamente agevole e autonoma oppure modificarla a piacimento in maniera da adattarla a specifiche esigenze.
Per motivi di opportunità eviteremo di realizzare il PCB e il montaggio del circuito limitandoci, come al solito, ad utilizzare la scheda di sviluppo EasyPIC3, ovviamente configurata in maniera opportuna. A tale proposito l'impostazione di jumper e dip-switch verrà eseguita in maniera tale da riprodurre esattamente il circuito elettrico di riferimento (Figura 1). La stessa applicazione potrà essere facilmente testata attraverso l'impiego della stessa EasyPIC3 o di altre demo board o schede di sviluppo analoghe oppure simulata attraverso il simulatore Real PIC Simulator.
L'impostazione dell'applicazione ricalca in buona parte quella presentata nell'articolo "*Orologio digitale con display a sette segmenti*" apparso sul numero *307 di Fare Elettronica* (*Autore A. Giannico*) con la sostanziale differenza di essere, per ovvi motivi, implementata in codice MikroPascal PRO for PIC invece che in codice C CCS e di presentare alcuni miglioramenti implementativi. Chi è interessato e dispone del numero 307 di Fare Elettronica potrà approfittarne per effettuare untile interessante confronto critico tra i due codici di programma.

IMPOSTAZIONE DEL PROGETTO E FUNZIONALITÀ DELL'APPLICAZIONE

Quando si sviluppa un'applicazione firmware un po' più complessa del solito è opportuno eseguire a priori una analisi che definisca in maniera precisa e chiara quali sono le funzionalità che l'applicazione dovrà rendere disponibili e quali i blocchi di codice (funzioni e procedure) in cui è opportuno scomporre l'applicazione stessa per renderla oltre che pienamente funzionante anche e ben strutturata. E' ovviamente necessario definire anche con attenzione l'interfaccia che fisicamente consentirà all'utente di interagire con l'appli-

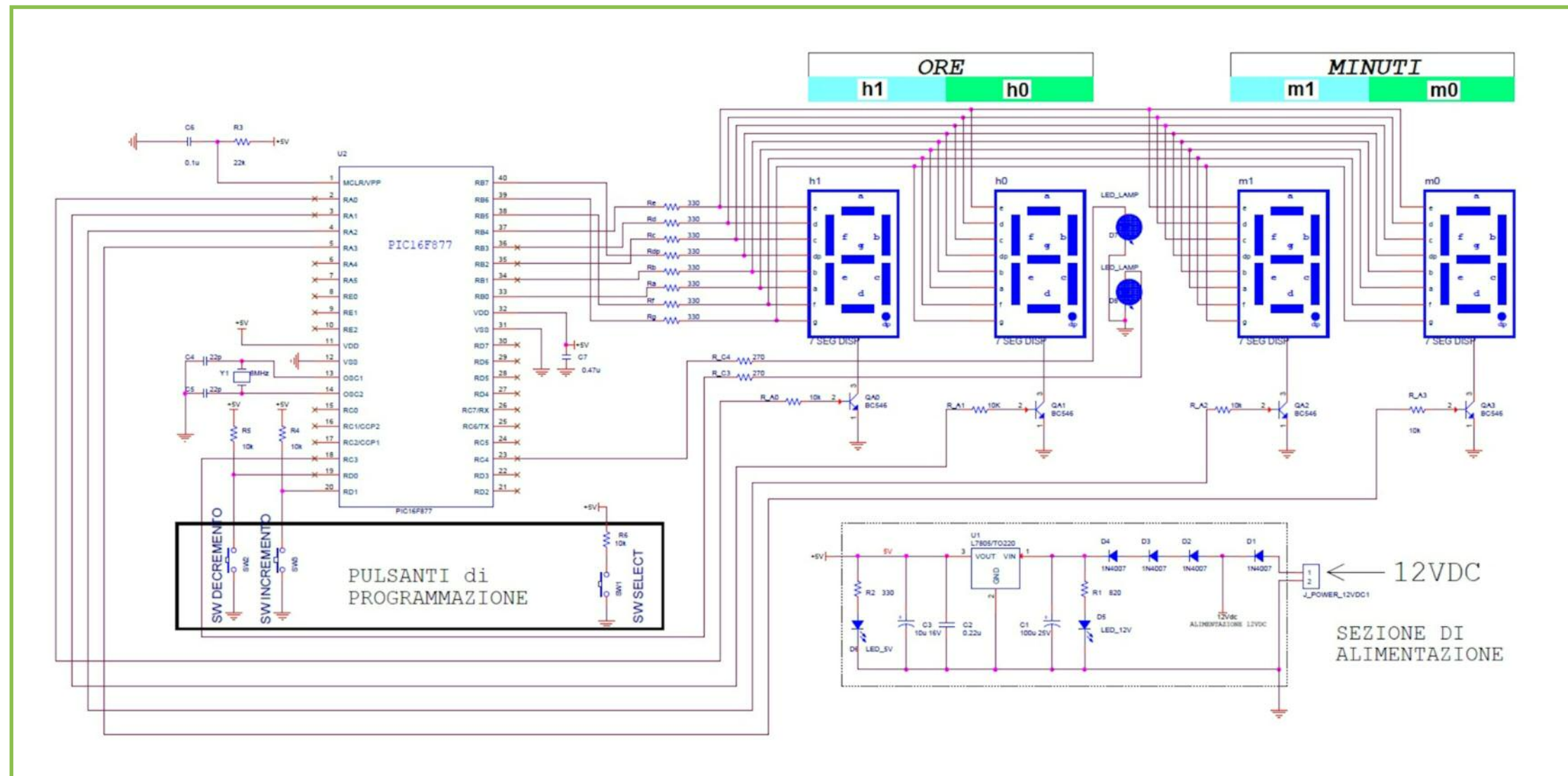



Figura 1-Schema elettrico dell'orologio digitale con PIC16F877 e display a 7 segmenti a 4 digit (rappresentazione oraria nella versione hh:mm). Aggiungendo altri due display abilitati rispettivamente dai pin RC0 ed RC1 è possibile estendere il circuito alla visualizzazione hh:mm:ss senza dover apportare alcuna modifica al firmware (per il quale si rimanda alla figura 3)

cazione. Quest'ultimo punto tende a definire la parte hardware e quindi circuitale dell'applicazione a differenza dei precedenti che tendono a definire una corretta strutturazione del codice.
Tale strutturazione è spesso facilitata dalla stesura di un diagramma a stati che descriva completamente e in modo chiaro le funzionalità dell'applicazione, gli stati in cui la stessa può trovarsi in un generico istante t e le modalità di transizione da uno stato all'altro in funzione delle interazioni con il mondo esterno.
Prima del codice dovrà essere il diagramma degli stati a rispondere, pienamente dal punto di vista funzionale, alla specifica esigenza applicativa. Un diagramma a stati lineare, ben strutturato e leggibile è la basa per la stesura di un codice firmware altrettanto lineare e comprensibile non solo al programmatore in fase di sviluppo ma anche a eventuali manutentori futuri del firmware.
Il firmware implementa l'orologio nel formato *hh:mm:ss* ma tale visualizzazione sarà realmente sfruttata o meno a seconda che si utilizzino 6 display a 7 segmenti comandati dal controllore oppure unicamente quattro display a 7 segmenti come effettivamente disponibile sulla scheda di sviluppo Easy PIC 3 e come riportato nello schema elettrico di figura 1. In questo caso il conteggio dei secondi avverrà comunque in background anche se non sarà visualizzato.
La limitazione in fase di test sarà in questo caso superata grazie all'impiego del simulatore Real PIC Simulator. In ogni caso, come vedremo, sarà possibile programmare l'orologio, cioè impostare manualmente ore, minuti e secondi anche disponendo di soli quattro display a 7 segmenti (destinati alla rappresentazione di ore e minuti).
In altri termini, ebbene l'applicazione firmware preveda l'impiego di sei display a sette segmenti, l'implementazione hardware e il test su EasyPIC3 può essere limitata anche alla visualizzazione delle sole cifre *hh:mm*.
Il firmware sviluppato potrà cioè essere comunque impiegato indifferentemente e senza modifiche tanto per la visualizzazione dell'orario nel formato ridotto "ore, minuti" (*hh:mm*) quanto per quella nel formato esteso "ore, minuti, secondi" (*hh:mm:ss*).
Prima di passare all'implementazione hardware e firmware dell'applicazione, specifichiamo i dati del problema e le funzionalità dell'applicazione:

- l'orologio dovrà implementare il formato *hh:mm:ss* (ore, minuti, secondi);
- la visualizzazione dell'orario deve essere effettuata su display a sette segmenti pilotati in multplexing;
- l'orologio dovrà essere programmabile cioè impostabile manualmente al fine di consentire, all'occorrenza, la correzione dell'ora;
- all'avvio l'orologio dovrà presentarsi nello stato orario 00:00:00;
- dovrà prevedere due modalità operative: una di visualizzazione e una di programmazione gestibili attraverso opportuni pulsanti;
- la modalità di visualizzazione dovrà consentire la visualizzazione dell'ora corrente nel formato "hh:mm" nel caso in cui si impieghino quattro soli display a 7 segmenti e nel formato "hh:mm:ss" nel caso in cui si impieghino 6 display 7 segmenti;
- la modalità di programmazione deve consentire il passaggio, mediante l'impiego di un apposito pulsante di selezione, attraverso i seguenti stati: visualizzazione, correzione secondi, correzione minuti, correzione ore;
- La permanenza nello stato di correzione secondi deve consentire l'azzeramento

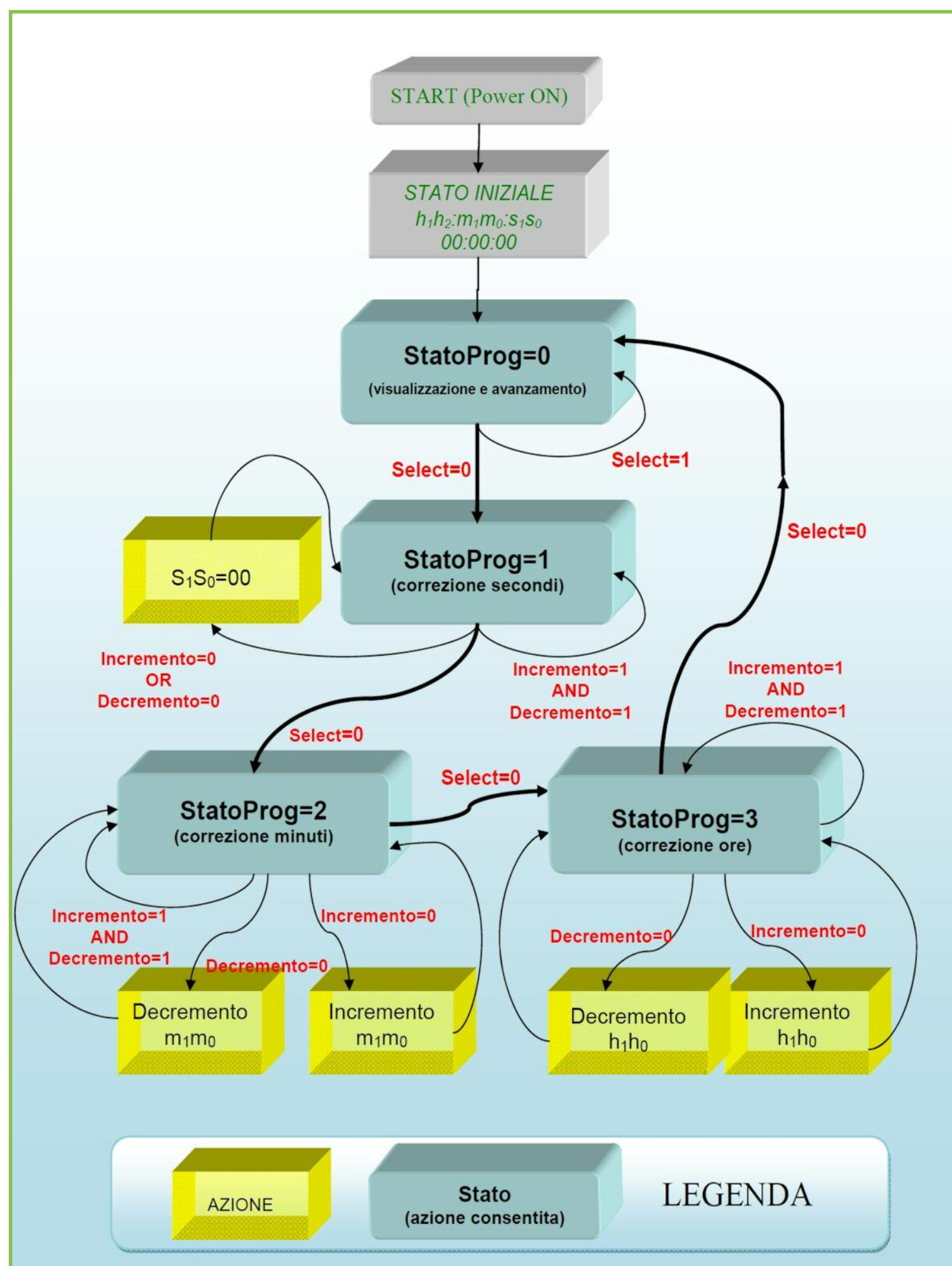



Figura 2-Diagramma degli stati e modalità di transizione da uno stato all'altro. StatoProg=0 indica che l'applicazione è in stato di visualizzazione. StatoProg > 0 indica che l'applicazione è in uno stato di programmazione cioè di correzione dell'ora (StatoProg=1àcorrezione secondi; StatoProg=2àcorrezione minuti; StatoProg=3àcorrezione Ore)

del campo secondi mediante l'impiego dei pulsanti utilizzati per l'incremento o il decremento di minuti e ore;

- la permanenza nello stato di correzione minuti deve consentire la correzione dei minuti mediante gli stessi pulsanti di incremento e decremento;
- la permanenza nello stato di correzione ore deve consentire, attraverso gli stessi pulsanti di incremento e decremento la modifica del campo ore;
- durante la programmazione del campo secondi sul display deve comparire il solo campo secondi (oppure i due punti lampeggianti di separazione tra ore e minuti, nel caso in cui il display disponibile sia a sole quattro cifre);
- durante la programmazione del campo minuti deve essere visualizzato il solo campo minuti;
- durante la programmazione del campo ore deve essere visualizzata il solo campo ore.

Il rispetto delle specifiche riportate negli ultimi tre punti consentono funzionalmente all'operatore di comprendere immediatamente e in maniera intuitiva se l'orologio è, in un dato istante, in uno stato di visualizzazione o di programmazione e nel secondo caso quale sia il campo orario modificabile in quel momento attraverso i pulsanti di incremento e decremento.

In conclusione, mentre i pulsanti DECREMENTO e INCREMENTO consentono di modificare i campi "secondi", "minuti" e "ore" imprimendo a questi ultimi rispettivamente un decremento o un incremento nel rispettivo campo di variazione, il pulsante di selezione, che d'ora in avanti chiameremo SELECT, dovrà consentire il passaggio dallo stato di visualizzazione a quello di programmazione (cioè correzione) e in questo ambito il passaggio da uno stato di programmazione all'altro (…àstato di visualizzazioneàcorrezione secondiàcorrezione minutiàcorrezione oreàritorno allo stato di visualizzazioneà…) fino al ritorno allo stato di visualizzazione. Il passaggio attraverso questi stati deve avvenire cioè, per ovvi motivi funzionali, in maniera ciclica. Sulla base di questi dati è possibile eseguire nell'ordine:

l'implementazione hardware dell'applicazione;

l'implementazione del diagramma a stati;

l'implementazione del firmware.

IMPLEMENTAZIONE HARDWARE: ASPETTI CIRCUITALI, DISPOSITIVI E FUNZIONALITÀ ASSOCIATE

Dalle specifiche consegue un hardware composto da:

- un microcontrollore PIC16F877;
- 4 o 6 display a 7 segmenti per la visualizzazione;
- 3 pulsanti di selezione/controllo.

Questi ultimi, in linea con i rispettivi significati, vengono indicati nello schema elettrico di Figura 1 con i seguenti nomi:

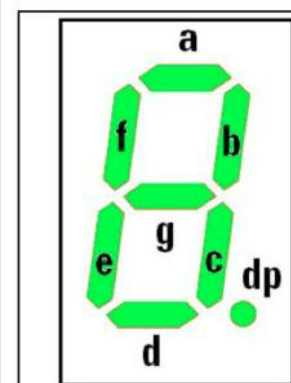
- SW SELECT: pulsante che consente di passare dalla modalità di visualizzazione che si ha di default all'accensione a quella di programmazione cioè correzione nell'ordine di secondi, minuti e ore; premendo ripetutamente il pulsante si passa sequenzialmente e in ordine ciclico attraverso gli stati riportati in Tabella 2: …à StatoProg=0 (Visualizzazione) à StatoProg=1 (correzione secondi) à Sta-

Segmento	Posizione
a	orizzontale superiore
b	verticale superiore destro
c	verticale inferiore destro
d	orizzontale inferiore
e	verticale inferiore sinistro
f	verticale superiore sinistro
g	orizzontale centrale
dp	punto decimale

Tabella 1-Nomenclatura e disposizione dei segmenti Led in un display a 7 segmenti a catodo comune

toProg=2 (correzione minuti) à StatoProg=3 (correzione ore) à StatoProg=0 (Visualizzazione)à…;

- SW INCREMENTO: pulsante la cui pressione determina l'incremento del valore del campo selezionato (ore o minuti); nel caso in cui il campo selezionato corrisponda a quello dei secondi, la pressione determina l'azzeramento del campo, esattamente come accade nei comuni orologi con display digitale. L'incremento è ciclico, pertanto se si sta agendo sui minuti la visualizzazione può raggiunge al massimo il valore 59 (mentre i campi hh ed ss sono oscurati), superato il quale riparte da "00"; cosa analoga dicasi per le ore con l'unica differenza che in questo caso l'incremento raggiunge il massimo valore "23" cui segue in maniera ciclica il valore "00";

Stato	Funzione principale	Descrizione dettagliata
StatoProg=0	Visualizzazione hh:mm:ss	-StatoTimer=1; -Visualizzazione Oraria; -Lampeggio Doppio Led.
StatoProg=1	Correzione secondi	-Visualizzazione del solo campo secondi s1s0; -StatoTimer=1 (fino a che non si preme il pulsante "Decremento" o "Incremento"); -Lampeggio doppio Led (fino a che non si preme il pulsante "Decremento" o "Incremento"); -Azzeramento S0 ed S1, interruzione del lampeggio doppio Led, StatoTimer=0 appena viene premuto il pulsante Incremento o Decremento.
StatoProg=2	Correzione minuti	-Visualizzazione del solo campo minuti (m1m0); -StatoTimer=1 se non è stato premuto il pulsante "Decremento" o "Incremento" nel precedente stato 1 e se gli stessi pulsanti non vengono premuti nemmeno nel presente stato; in caso opposto StatoTimer=0; -Lampeggio doppio Led se StatoTimer=1; -Pressione pulsante Decremento → Decremento minuti m1m0; -Pressione pulsante Incremento → Incremento minuti m1m0.
StatoProg=3	Correzione ore	-Visualizzazione del solo campo ore (h1h0); -StatoTimer=1 se non è stato premuto il pulsante "Decremento" o "Incremento" nei precedenti stati 1 e 2 e se gli stessi pulsanti non vengono premuti nemmeno nel presente stato; in caso opposto StatoTimer=0; -Lampeggio doppio Led se StatoTimer=1; -Pressione pulsante Decremento → Decremento ore h1h0; -Pressione pulsante Incremento → Incremento ore h1h0.

Tabella 2-Stati che caratterizzano l'applicazione e relativo significato (StatoProg=0 indica che l'applicazione non è in stato di programmazione mentre StatoProg>0 indica che l'applicazione è in uno stato di programmazione cioè di correzione di secondi, minuti oppure ore)

- SW DECREMENTO: agisce esattamente come il pulsante precedente con la differenza che la modifica del valore del campo selezionato attraverso il pulsante SELECT avviene ciclicamente per decremento; il decremento per i minuti può arrivare fino al valore "00" cui segue in modo ciclico il valore "59", nel caso in cui il campo selezionato sia costituito dai minuti, oppure "23" nel caso in cui il campo selezionato sia costituito dalle ore.

Nel caso in cui il campo selezionato sia costituito dai secondi, come già detto, sia la pressione del pulsante SW INCREMENTO che quella del pulsante SW DECREMENTO determinano l'azzeramento del campo.
Il diagramma a stati di Figura 2 esprime efficacemente la logica di funzionamento fin qui descritta.
Occorre, per una corretta comprensione, sottolineare alcuni aspetti circuitali dell'applicazione. In particolare, è opportuno osservare che la pressione del pulsante SW SELECT che consente la transizione da uno stato all'altro corrisponde alla transizione logica che porta l'ingresso sul PIC (pin RD3) dallo stato logico 1 allo stato logico 0 essendo il pulsante connesso in pull-up (schema elettrico di Figura1). Cosa analoga vale per i pulsanti SW DECREMENTO e SW INCREMENTO.
Lo schema elettrico riportato in Figura 1 comprende, oltre ala sezione di programmazione e visualizzazione, una sezione di alimentazione. L'alimentazione 5Vdc, utile ad alimentare il controllore, è ricavata dalla 12Vdc attraverso un regolatore 7805 e alcuni diodi di limitazione che servono a limitare la tensione in ingresso al regolatore in maniera da limitarne il drop-out e quindi la potenza termica dissipata sullo stesso regolatore. Questo semplice accorgimento consente di limitarne il riscaldamento e di evitare, la necessità di munire il regolatore di un dissipatore termico.
Due led, rispettivamente in ingresso e in uscita allo stadio, consentono di controllare lo stato dell'alimentazione in ingresso 12Vdc e di quella in uscita 5Vdc.

IL PILOTAGGIO DEI DISPLAY A 7 SEGMENTI

Per il pilotaggio dei display a sette segmenti vale quanto detto nelle precedenti lezioni sul controllo in multiplexing. A tale proposto, come al solito, nel dimensionamento del circuito e nella stesura del firmware va tenuto conto del fatto che la luminosità dei singoli segmenti componenti i display non è solo inversamente proporzionale alle resistenze di limitazione ma anche al numero di display pilotati. Ci limitiamo a parte questo a ricordare (Tabella 1) la tipica struttura del display a sette segmenti utilizzato nell'applicazione.

IL FLUSSO DI PROGRAMMA (DIAGRAMMA A STATI)

Con riferimento al diagramma a stati riportato in Figura 2, si nota che all'avvio l'applicazione si trova nello stato di visualizzazione dell'orario (*StatoProg=0*) con stato orario corrente *hh:mm:ss=00:00:00*.
Nel caso in cui non venga premuto alcun pulsante il tempo comincia a scorrere producendo l'avanzamento della visualizzazione.
In divere occasioni e in particolar modo all'avvio può essere necessario correggere

Campo del Display	Significato dei digit
Campo Ore (h1h0)	h1: cifra delle decine h0: cifra delle unità
Campo Minuti (m1m0)	m1: cifra delle decine

Tabella 3-Significato della variabile StatoTimer

	m0: cifra delle unità
Campo Secondi (S1S0)	S1: cifra delle decine S0: cifra delle unità

Tabella 4- Campi del display e significato dei relativi digit

il valore corrente dell'orario. Alla prima pressione del pulsante SELECT la variabile "Select" si porta da livello alto (5V) a livello basso (si noti il pull-up sul pin RD3 del controllore-Figura 1) e questo porta l'applicazione nella modalità (Stato) di programmazione (correzione) secondi (*StatoProg=1*). Con la stessa logica, premendo nuovamente lo stesso pulsante si passa nello stato di programmazione (correzione) minuti (*StatoProg=2*) e premendolo una terza volta si entra nella stato di programmazione (correzione) ore (StatoProg=3).

Nel primo caso (*StatoProg=1*) viene visualizzato il solo campo secondi sul display, nel secondo (*StatoProg=2*) il solo campo minuti e nel terzo (*StatoProg=3*) il solo campo ore. In questo modo appare del tutto intuitivo, quale sia, in ciascuno stato (*StatoProg=1,2,3*), il campo orario (secondi, minuti o ore) abilitato alla correzione. Entrati nella modalità di programmazione (*StatoProg=1,2,3*) attraverso la pressione del pulsante SELECT, il conteggio temporale non è interrotto ma continua ad essere attivo. Di ciò tiene conto la variabile *StatoTimer* il cui valore permane a 1, anche se sul display è visualizzato un solo campo (secondi, minuti oppure ore-si vedano le Figure 6, 7, 8, 10, 11, 12, 13 per dettagli); lo scorrere del tempo continua in altre parole ad essere attivo in background. Solo nel caso in cui durante la fase di programmazione venga premuto almeno una volta il pulsante SW INCREMENTO o il pulsante SW DECREMENTO si ha l'interruzione, cioè il congelamento, del Timer. Infatti, solo nel caso in cui si verifichi questo evento è inequivocabile il fatto che l'utente voglia realmente reimpostare l'orario.

Nel caso in cui si passa invece tra due o più stati di programmazione (*StatoProg=…à1à2à3à..*), premendo ripetutamente il pulsante SELECT senza tuttavia agire mai sui pulsanti SW INCREMENTO o SW DECREMENTO, il conteggio del Timer continua in background (la visualizzazione proposta in questi frangenti riguarda però unicamente secondi, minuti oppure ore a seconda che si è rispettivamente nello stato di programmazione (StatoProg) 1, 2 oppure 3.

Una volta entrati nella modalità di programmazione (*StatoProg=1*), per ritornare alla modalità di visualizzazione completa (*StatoProg=0*) è necessario attraversare

1-Orologio basato sull'evoluzione del Timer0
2-Pilotaggio dei display

```
{*
 * Corso: MikroPascal PRO for PIC-Fare Elettronica-Lezione n.: 10
 * Nome progetto: OrologioProgrammabile-Rev.0: 2013-05-11; Autore: A. Giannico
 * Descrizione: Il codice di programma implementa un orologio programmabile
   h1h0:m1m0:s1s0 dove h1: ore (cifra delle decine); h0: ore (cifra delle unità)
                       m1: minuti (c. delle decine); m0: minuti(c. delle unità)
                       s1: secondi(c. delle decine); s0:secondi (c. delle unità)
   L'orario nel formato ore: minuti: secondi viene rappresentato su display a
   sette segmenti. L'assegnazione dei pin è fatta in maniera da rendere il
   codice compatibile con EasyPIC3. Vengono utilzzate le nozioni apprese nelle
   precedenti lezioni, in particolare l'uso del Timer0 e quello dei display a
   sette segmenti pilotati in multplexing.
 * MCU:PIC18F877; Sch. di sviluppo:Easypic3; Oscillatore: HS 8.0000 MHz
 * SW: mikroPascal PRO for PIC  ver. 5.61
   http://www.mikroe.com/eng/products/view/10/mikropascal-pro-for-pic/
 *}
//____________________________________________________________
program OrologioProgrammabile;
var StatoTimer, errore: Boolean;
var StatoProg:byte;
var S1, S0, m1, m0, h1, h0: byte;
var count_OF:integer;
const DURATA_SECONDO=7812;
const TMUX=300;

procedure interrupt(); //__________Interrupt Timer0 --> aggiornamento hh:mm:ss
begin
if (TestBit(INTCON, TMR0IF) = 1) then
      begin
      ClearBit(INTCON, TMR0IF);
      count_OF:=count_OF+1;
      //______________________Gestione dell'errore di overflow del timer0
      if ((errore=1)AND(count_OF = DURATA_SECONDO)AND(StatoTimer=1)) then
                                                begin
                                                count_OF:=count_OF-1;
                                                errore:=0;
                                                end
                                                else
         if ((errore=0)AND(count_OF = DURATA_SECONDO)AND(StatoTimer=1)) then
                                                       begin
                                                       errore:=1;
                                                       end;
      //____________________________________________________
      if ((count_OF = DURATA_SECONDO)AND(StatoTimer=1)) then
                  begin
                  S0:=S0+1;                               //scatta il secondo
                  portc.3:=not(portc.3);
                  portc.4:=not(portc.4);
                  if (S0=10) then
                          begin
                          S0:=0;
                          S1:=S1+1;
                          end;
                  if (S1=6)  then
                          begin
                          S1:=0;
                          m0:=m0+1;
                          end;
                  if (m0=10) then
                          begin
                          m0:=0;
                          m1:=m1+1;
                          end;
                  if (m1=6)  then
                          begin
                          m1:=0;
                          h0:=h0+1;
                          end;
                  if ((h0=10)AND(h1<2)) then
                                 begin
                                 h0:=0;
                                 h1:=h1+1;
                                 end;
                  if ((h1=2)AND(h0=4)) then
                                 begin
                                 h0:=0;
                                 h1:=0;
                                 end;
                  count_OF:=0;
                  end;
      end;
end;

procedure Display7seg(dato:integer); //___Aggiornamento cifra del Display 7 seg
begin
    case (dato) of
               0: portB:=0x3F;
               1: portB:=0x06;
               2: portB:=0x5B;
               3: portB:=0x4F;
               4: portB:=0x66;
               5: portB:=0x6D;
               6: portB:=0x7D;
               7: portB:=0x07;
               8: portB:=0x7F;
               9: portB:=0x6F;
               end;
end;

procedure Display7segOFF();          //_____Procedura di spegnimento del display
begin                                //_____precede lo step di multiplexing
 portB:=0x00;
 portA:=0x00;
end;

begin               // ___________________________CICLO PRINCIPALE di programma
delay_ms(1000);
INTCON.GIE  := 1;   // GIE: Global Interrupt Enable bit
INTCON.PEIE := 0;   // PEIE: Peripheral Interrupt Enable bit
INTCON.TMR0IE := 1;// TMR0 Interrupt Enable
OPTION_REG.PSA:=1; // prescaler non applicato
OPTION_REG.T0CS:=0;// Sorgente per TMR0: Fosc/4
OPTION_REG.T0SE:=0;// Timer0 Source Edge (su fronte di salita)
trisA:=0x00;        // impostazione registro Tris_a     --> porta A uscita
trisB:=0x00;        // impostazione registro Tris_b     --> porta B uscita
trisC:=0x00;        // impostazione del registro Tris C --> porta C uscita
trisD:=0xFF;        // impostazione del registro Tris D --> porta D ingresso
S1:=0; S0:=0;       // l'orologio parte per fefault da 00:00:00
m1:=0; m0:=0;
h1:=0; h0:=0;
StatoTimer:=1;      // l'orologio parte da stato=1 (attivo cioè in avanzamento)
errore:=0;          // l'errore di conteggio accumulato è inizialmente nullo
StatoProg:=0;       // l'orologio non è inizialmente in stato di programmazione
TMR0 := 0;          // il registro TMR0 (Timer0) parte da valore nullo
count_OF:=0;
while (TRUE) do
       Begin
       if(((PortD.3)=0)AND(StatoProg=0)) then
               begin
               StatoProg:=1;
               while ((PortD.3)=0) do
                                begin
                                end;
               end;
       if(((PortD.0=0)OR(PortD.1=0))AND(StatoProg=1)) then
               begin
               StatoTimer:=0;
               S0:=0; S1:=0;
               end;
       if((portD.3=0)AND(StatoProg=1)) then
               begin
               StatoProg:=2;
               while (PortD.3=0) do
                              begin
                              end;
               end;
       if((PortD.0=0)AND(StatoProg=2)) then
               begin
               StatoTimer:=0;
               if ((m0>=1)AND(m0<=9)) then m0:=m0-1
                                      else
                                          begin
                                          m0:=9;
                                          if (m1>0) then m1:=m1-1
                                                    else m1:=5;
                                          end;
               while ((PortD.0=0))  do
                                 begin
                                 end;
               end;
       if(((PortD.1=0))AND(StatoProg=2))  then
               begin
               StatoTimer:=0;
               if ((m0<=8)AND(m0>=0)) then m0:=m0+1
                                      else
                                          begin
                                          m0:=0;
                                          if (m1<5) then m1:=m1+1
                                                    else m1:=0;
                                          end;
               while ((portD.1=0)) do
                                begin
                                end;
               end;
       if(((PortD.3)=0)AND(StatoProg=2)) then
               begin
               StatoProg:=3;
               while ((PortD.3)=0) do
                                 begin
                                 end;
               end;
       if(((PortD.0=0))AND(StatoProg=3)) then
               begin
               StatoTimer:=0;
               if (h0>=1) then h0:=h0-1
                          else
                          begin
                          if (h1=2) then
                                  begin
                                  h1:=1; h0:=9;
                                  end
                                  else if (h1=1) then
                                                 begin
                                                 h1:=0; h0:=9;
                                                 end
                                                 else if (h1=0) then
                                                                begin
                                                                h1:=2;h0:=3;
                                                                end;
                          end;
                              while ((PortD.0=0))  do
                                                   begin
                                                   end;
               end;
       if(((portD.1=0))AND(StatoProg=3)) then
               begin
               StatoTimer:=0;
               if ((h1<2)AND(h0<=8)) then h0:=h0+1
                          else if ((h1<2)AND(h0=9)) then
                                  begin
                                  h0:=0; h1:=h1+1;
                                  end
                                  else if ((h1=2)AND(h0<3)) then
                                                           begin
                                                           h0:=h0+1;
                                                           end
                                  else if ((h1=2)AND(h0=3)) then
                                                           begin
                                                           h0:=0; h1:=0;
                                                           end;
               while ((portD.1=0)) do
                                 begin
                                 end;
               end;
       if(((PortD.3=0))AND(StatoProg=3)) then
               begin
               StatoProg:=0;
               StatoTimer:=1;
               while ((PortD.3=0)) do
                                 begin
                                 end;
               end;
       //__________________________________________________________display h0
       Display7segOFF();
       PortC.0:=0;
       PortC.1:=0;
       if ((StatoProg=0)OR(StatoProg=3)) then Display7seg(h0);
       PortA:=0x02;
       delay_us(TMUX);
       //__________________________________________________________display h1
       Display7segOFF();
       if ((StatoProg=0)OR(StatoProg=3)) then Display7seg(h1);
       portA:=0x01;
       delay_us(TMUX);
       //__________________________________________________________display m0
       Display7segOFF();
       if ((StatoProg=0)OR(StatoProg=2)) then Display7seg(m0);
       portA:=0x08;
       delay_us(TMUX);
       //__________________________________________________________display m1
       Display7segOFF();
       if ((StatoProg=0)OR(StatoProg=2)) then Display7seg(m1);
       portA:=0x04;
       delay_us(TMUX);
       //__________________________________________________________display s0
       Display7segOFF();
       portC.0:=1;
       portC.1:=0;
       if ((StatoProg=0)OR(StatoProg=1)) then Display7seg(s0);
       delay_us(TMUX);
       //__________________________________________________________display s1
       Display7segOFF();
       portC.0:=0;
       portC.1:=1;
       if ((StatoProg=0)OR(StatoProg=1)) then Display7seg(s1);
       end;
end.
```

3-Macchina a stati
4-Multiplexing dei display

Figura 3-Codice MikroPascal dell'applicazione; si notino la sezione di temporizzazione a Timer (1), quella di controllo del dei display a sette segmenti, quella che implementa la macchina a stati finiti di Figura 2 e quella di multiplexing dei display.

tutti gli stati di programmazione (transizione *StatoProg=1àStatoProg=2àStatoProg=3*) fino a tornare in quello zero (transizione *StatoProg=3àStatoProg=0*) sia che si apportino modifiche all'orario sia che non se ne apportino.
Quanto espresso è sintetizzato efficacemente nel diagramma a stati di Figura 2 e in Tabella 2 dove sono riportati in dettaglio i significati degli stati rappresentati all'interno del diagramma.
I due punti lampeggianti sono rappresentati dai led posti rispettivamente sui pin RC3 e RC4 del controllore. Essi rappresentano i "due punti" lampeggianti che normalmente separano, in un orologio digitale, la visualizzazione del campo ore da quella del campo minuti. Quanto esposto si completa con le informazioni riassunte in Tabella 3 che riportano il significato, peraltro ovvio, della variabile *StatoTimer* (stato "attivo" o di "congelamento" del timing). Si rimanda il lettore a un'attenta analisi del diagramma a stati di Figura 2 e delle tabelle 2 e 3 prima di passare all'analisi diretta del codice MikroPascal discusso nei paragrafi che seguono e riportato in Figura 3. Le variabili *StatoProg* e *StatoTimer*, i cui significati sono riportati rispettivamente in Tabella 2 e Tabella 3, sono ovviamente utilizzate con il medesimo significato all'interno del diagramma di stato (Figura 2) e all'interno del listato Firmware (Figura 3).

IL FIRMWARE

Il Firmware (Figura 3) rispecchia quanto fin qui detto sull'implementazione hardware e sull'impiego dei pulsanti di programmazione. Si nota in particolare:

- il pin di incremento (*RD1*) e quello di de-

Riga di codice	Significato
INTCON.GIE := 1;	GIE: Global Interrupt Enable bit
INTCON.PEIE := 0;	PEIE: Peripheral Interrupt Enable bit
INTCON.TMR0IE := 1;	TMR0 Interrupt Enable
OPTION_REG.PSA:=1;	prescaler non applicato
OPTION_REG.T0CS:=0;	Sorgente per TMR0: Fosc/4
OPTION_REG.T0SE:=0;	Timer0 Source Edge (fronte di salita)
TMR0:= 0;	il registro TMR0 parte da valore nullo

Tabella 5-Righe di codice di impostazione del Timer0 e relativo significato. Per una maggiore comprensione si faccia anche riferimento al diagramma a blocchi del Timer0 riportato nelle precedenti lezioni 8 e 9 cui si rimanda per dettagli.

cremento (*RD0*) dei valori dei campi hh, mm ed ss; su tali pin sono connessi i pulsanti omonimi SW INCREMENTO e SW DECREMENTO;

- il pin RD3 su cui insiste il pulsante di selezione (SELECT) della modalità di programmazione (secondi, ore, minuti) e di ripristino della modalità di visualizzazione;
- la porta di pilotaggio in multiplexing dei display 7 segmenti (porta B) e quella di pilotaggio dei transistor di attivazione dei catodi (porta A e pin RC0 ed RC1);
- i pin di pilotaggio dei led che rappresentano il doppio punto di separazione tra ore e minuti (RC3 ed RC4).

I secondi, i minuti e le ore sono rappresentate, all'interno del listato, dalle corrispondenti variabili riportate in Tabella 4. Il funzionamento firmware dell'applicazione ruota intorno all'evoluzione del Timer 0 che a sua volta presenta come sorgente di clock Fosc/4 dove Fosc è la frequenza di clock di sistema imposta dal solito quarzo ad 8MHz (valore tipico del quarzo in dotazione sulla scheda di test Easy PIC 3).
All'inizio del Main Program viene inizializzato l'interrupt Timer0 (vedi listato di figura 3). La variabile *StatoTimer* con il suo valore indica se il conteggio del Timer è in corso (*StatoTimer=ON=1*) o "congelato" (*StatoTimer=OFF=0*) come riassunto all'interno della Tabella 3. All'interno del ciclo "While (TRUE)", individuato nel diagramma di Figura 2 dalle frecce (di transizione da uno stato all'altro) di tipo spesso, è gestito il flusso di programma. All'interno di questo ciclo viene gestita la visualizzazione (StatoProg=0) in multiplexing e la programmazione di secondi (*StatoProg=1*), minuti (*StatoProg=2*) e ore (*StatoProg=3*).
Il multiplexing, cioè la selezione della cifra da rappresentare di volta in volta e quindi del display da attivare (h1, h0, m1, m0, S1 o S0) attraverso il relativo transistor di pilotaggio è gestito direttamente all'interno del ciclo "While (TRUE)", mentre l'eccitazione dei segmenti del display selezionato è gestita mediante la procedura "Display7seg" (vedi listato in Figura 3) cui viene passata di volta in volta la cifra da rappresentare. In base al valore ricevuto la procedura individua i segmenti da eccitare e attiva il corretto pilotaggio della porta B.

IMPOSTAZIONE DEL TIMER

La precisione dell'orologio dipende dalla corretta impostazione del timer. Fondamentale importanza ricopre, se si vuole che l'orologio non accumuli né anticipi né ritardi di temporizzazione, la corretta impostazione del Timer0. Vediamo in dettaglio in che modo sono stati scelti i valori dei relativi parametri. A riguardo, il listato di Figura 3 riporta, in particolare, le righe di codice elencate in Tabella 5 con il relativo significato.
Altrettanto importante è il valore assegnato alla costante DURATA_SECONDO. Le due cose, insieme, determinano il numero di overflow del Timer0 necessari ad accumulare un tempo complessivo pari ad 1 secondo. Questo evento è infatti quello cui è "agganciato" l'aggiornamento del campo secondi dell'orologio e di conseguenza dei campi minuti e ore. Un errore nella definizione di questo parametro si ripercuote direttamente su un errore di temporizzazione. Vediamo in che modo è possibile definirlo correttamente.
Il Timer0 è un timer a 8 bit che va pertanto in overflow dopo essere passato per 256 stati. La frequenza di clock che ne determina l'evoluzione è Fosc/4 dove Fosc è la frequenza di sistema imposta attraverso il quarzo esterno connesso ai pin OSC1 ed OSC2 del controllore. Non essendo applicato alcun fattore di prescaler, un overflow si verifica dopo un tempo pari a:

$$T_{of}= 256xT_{clock}x4=256x(1/F_{osc})x4=256x(1/8MHz)x4=128\ us$$

Evidentemente, per coprire un tempo pari a 1 secondo è necessario un numero di overflow pari a:

$$N_{OF}=1s/128us=7812,5$$

Ovviamente, non è possibile eseguire un numero di overflow non intero per cui è necessario scegliere tra i valori 7812 e 7813. La scelta del valore 7812 comporta la perdita di 0,5 secondi ogni 7812,5 secondi cioè

(0,5/7812,5)x60x60x24=5,5 secondi al giorno

dove i due 60 sono rispettivamente i minuti in un'ora e i secondi in un minuto, 24 sono le ore in un giorno mentre 0,5/7812,5 e la frazione di tempo di cui si commette errore in mancanza di strategie di correzione dello stesso.
Analogamente, la scelta di 7813 porta a sopravanzare un orologio di riferimento dello stesso tempo.
Per ovviare a questo problema è necessario implementare, all'interno del firmware, un controllo che faccia durare un secondo una volta 7812 overflow e la successiva 7813 overflow in maniera alternata. In questo modo la durata media di un secondo è pari proprio a 7812,5 overflow e l'orologio non accumula né anticipi né ritardi. Questo correzione è implementata nella sezione del codice contrassegnata (Figura 1) con il commento "Gestione dell'errore di overflow del timer0". La variabile "errore" assume valore 1 quando il conteggio si ferma a 7812 a indicare che al prossimo ciclo è necessario accumulare un overflow in più per compensare l'errore. A questo

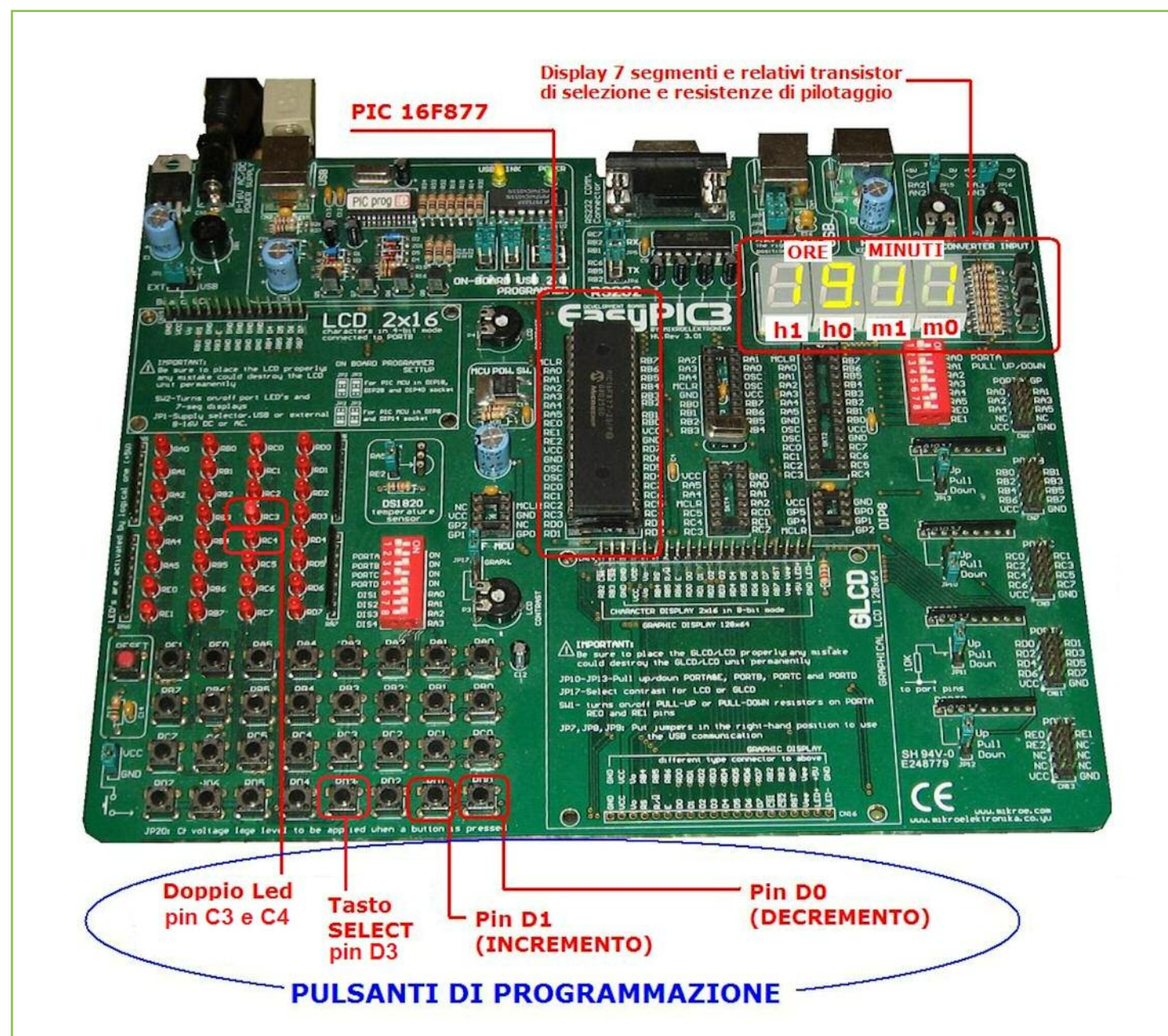



Figura 4-Scheda di sviluppo Easy PIC 3 e relative risorse hardware utilizzate per l'implementazione dell'applicazione e l'esecuzione dei test

punto la variabile viene riportata a valore 0 a indicare che il prossimo ciclo non necessita di correzione.
Per ulteriori dettagli si rimanda all'analisi approfondita del diagramma degli stati (Figura 2) e del listato di codice (Figura 3).

TEST DELL'APPLICAZIONE

Il test dell'applicazione è stato condotto attraverso la scheda di sviluppo Easy PIC 3 ma altre demo-board o schede di sviluppo sarebbero potute risultare parimenti valide allo scopo di implementare il circuito di Figura 1. Il test dell'applicazione è stato condotto verificando che lungo tutti i possibili percorsi individuabili all'interno del diagramma degli stati di Figura 2 la stessa rispondesse come atteso, sia in visualizzazione che in programmazione.
In Figura 4, sono indicati i display e i pulsanti utilizzati sulla Easy PIC 3 impiegata come test-board. Nelle Figure 5, 6, 7 e 8 è invece riportato la sequenza degli stati in cui l'orologio viene a trovarsi nel corso della visualizzazione e delle operazioni di programmazione. Il confronto della temporizzazione mantenuta dall'orologio con quella di un orologio di riferimento (per esempio il proprio orologio da polso o quello integrato nel sistema operativo del proprio PC) consente di verificare che il timing dell'applicazione non accumuli né anticipi né ritardi.

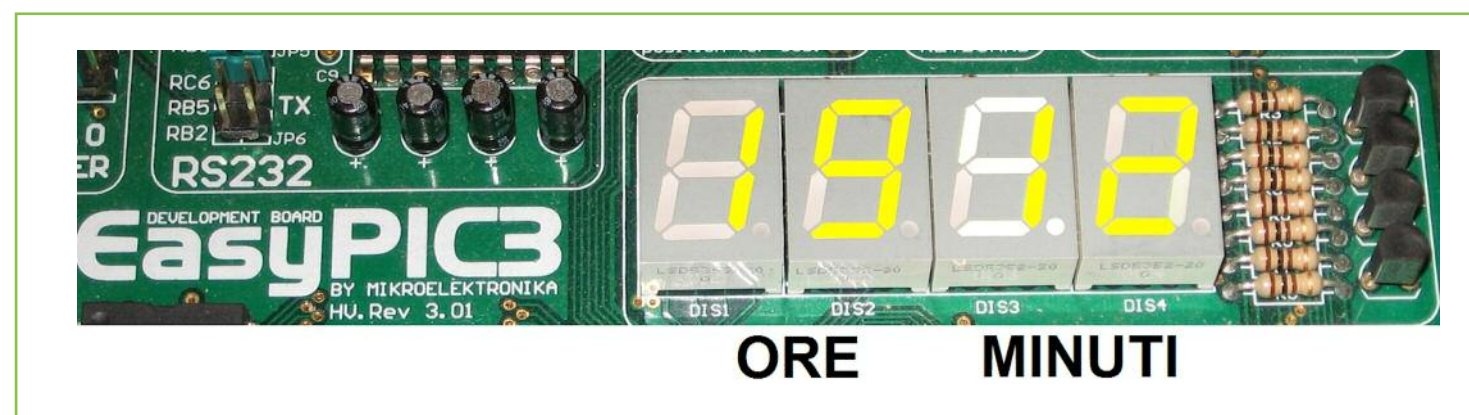



Figura 5-Display 4x7 segmenti della Scheda di sviluppo Easy PIC3 corrispondente allo schema di Figura 1 (a meno dei display del campo secondi) con applicazione nello stato StatoProg=0 (Stato di visualizzazione dell'orario)

Figura 6-Display 4x7segmenti della Scheda di sviluppo Easy PIC 3 corrispondente allo schema di Figura 1 con applicazione nello stato StatoProg=1 (Stato di programmazione azzeramento secondi); prima di operare l'azzeramento attraverso i pulsanti INCREMENTO o DECREMENTO i "due punti" rappresentati dai led C3 e C4 lampeggiano, segno che il Timer è attivo; ad azzeramento avvenuto essi smettono di lampeggiare e il Timer resta "congelato" fino a che non si esce completamente dal ciclo di programmazione).

Figura 7-Display 4x7segmenti della Scheda di sviluppo Easy PIC 3 corrispondente allo schema di Figura 1 con applicazione nello stato StatoProg=2 (Stato di programmazione dei minuti); attraverso i pulsanti INCREMENTO e DECREMENTO è possibile modificare il valore del campo "minuti"; attraverso il pulsante SELECT è possibile uscire da questa fase di programmazione e passare alla programmazione delle ore.

Figura 8-Display 4x7segmenti della Scheda di sviluppo Easy PIC 3 corrispondente allo schema di Figura 1 con applicazione nello stato StatoProg=3 (Stato di programmazione delle ore); attraverso i pulsanti INCREMENTO e DECREMENTO è possibile modificare il valore del campo "ore"; attraverso il pulsante SELECT è possibile uscire dallo stato di programmazione e tornare allo stato di visualizzazione.

Nelle Figure 9, 10, 11, 12 e 13 sono riportate alcune simulazioni del funzionamento dell'applicazione effettuate con il simulatore *Real PIC Simulator*. Si rimanda alla loro analisi per maggiori dettagli.

OSSERVAZIONI UTILI

Come è possibile osservare dall'analisi del listato non si è eliminato del tutto l'impiego di istruzioni di delay all'interno del blocco principale di codice e questo al fine di rendere semplice l'implementazione del multiplexing dei display a sette segmenti.

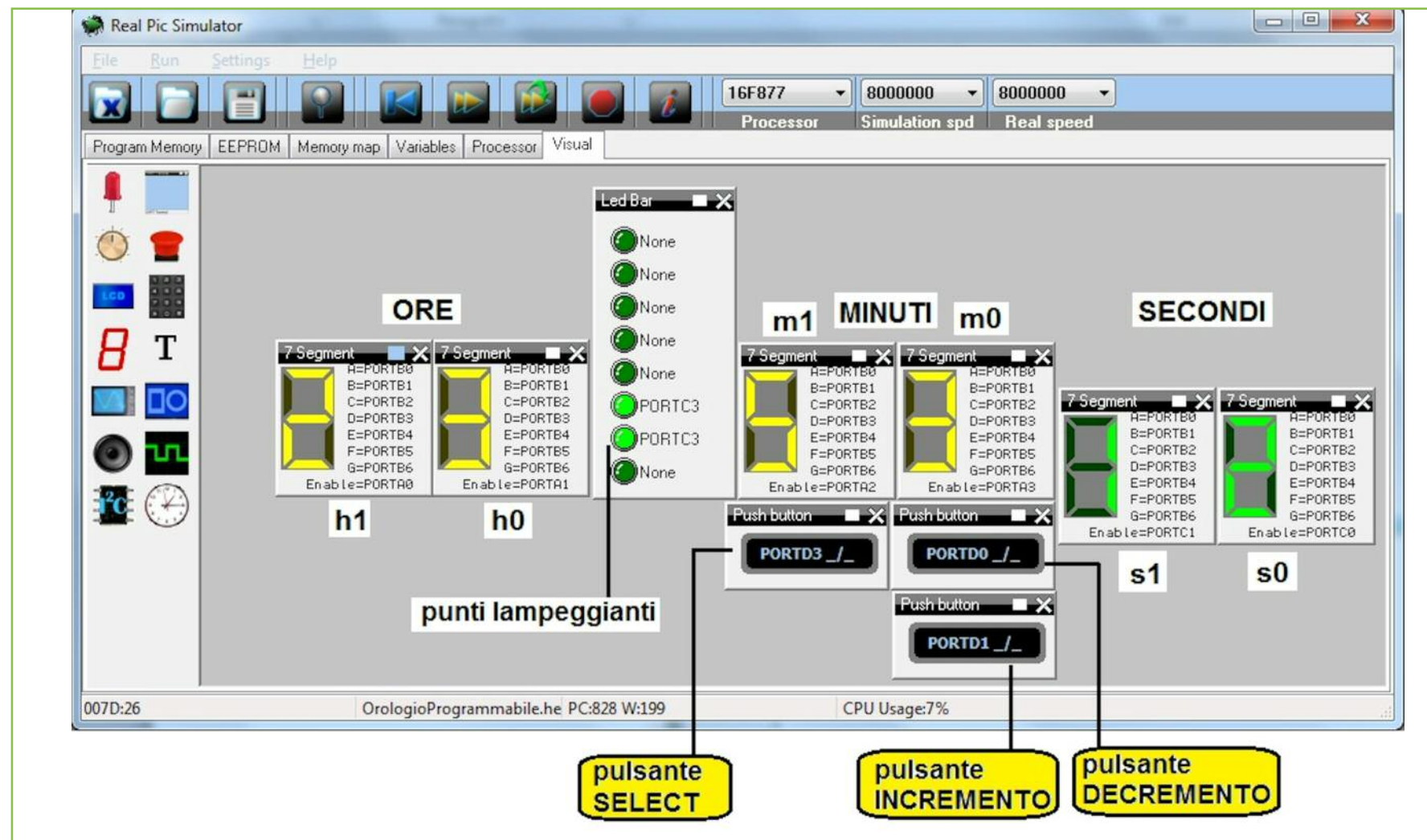



Figura 9-Simulazione dell'applicazione attraverso il simulatore Real PIC Simulator v.1.1.0.0 (è rappresentata l'applicazione nello stato di visualizzazione (StatoProg=0)

Ciascun display dei sei componenti il visualizzatore viene interessato per un tempo pari a 300 microsecondi prima di lasciare il passo al successivo. Ritardi di questo tipo nell'esecuzione del codice main non comportano particolari problemi all'intercettazione dei comandi provenienti dai tre pulsanti di programmazione. Infatti, 300 microsecondi per sei display assommano complessivamente a 1800 microsecondi cioè meno di 2 millisecondi, un tempo del tutto trascurabile rispetto a quelli che vengono coperti nella pressione di uno qualsiasi dei pulsanti impiegati. Inoltre, questi ritardi non comportano alcun problema alla corretta temporizzazione dal momento che la procedura di interrupt che gestisce l'evoluzione del tempo è basata sull'interrupt Timer0. Questo ci consente di evitare in l'impiego di pin ad interrupt RB On-Change che possono in tal modo essere utilizzati per il controllo dei display a sette segmenti.

Non abbiamo utilizzato la funzione Button per l'intercettazione dei pulsanti di programmazione. Dall'implementazione pratica dell'applicazione effettuata con Easy-PIC3 infatti si è potuto constatare come l'antirimbalzo non sia strettamente necessario. Coloro che intendono riprodurre l'applicazione possono in ogni caso provare a sostituire le istruzioni che intercettano la pressione dei pulsanti con l'istruzione Button.

Anche il lampeggio dei due punti è effettuato all'interno della procedura di interrupt ed è semplice verificare come essi continuino a lampeggiare regolarmente al frequenza di mezzo hertz (un accensione o uno spegnimento corrispondono ad un secondo di temporizzazione) anche mentre si premono i pulsanti. Essi si bloccano solo in occasione della programmazione dell'orologio nel momento in cui si agisce sull'impostazione dei secondi. L'impiego di Real PIC Simulator consente di verificare in maniera approfondita il funzionamento dell'applicazione anche se non si dispone della relativa implementazione fisica. Non è difficile modificare gli esempi mostrati in maniera da adattarli a scopi specifici.

Per quanto concerne l'impiego dei Timer abbiamo preso a riferimento il Timer0, ma l'implementazione di applicazioni firmware basate su Timer1 e Timer2 non è molto differente.

Si consiglia, a conclusione della lezione di:

- analizzare attentamente i listati di codice presentati alla luce di quanto riportato in figure e tabelle;
- riprodurre l'applicazione in hardware e firmware o solo in firmware attraverso l'impiego di simulatori come Real PIC Simulator.

Tra le piccole novità di cui è disseminato il codice MikroPascal dell'applicazione presentata in questa lezione segnaliamo l'operatore "not" utilizzato per ottenere la negazione dello stato di un bit di una porta precedentemente definita come porta di uscita. L'istruzione:

portc.3:=not(portc.3);

per esempio non fa altro che invertire lo stato logico del bit 3 della porta di uscita C.

UTILI VARIANTI: CRONOMETRO E BROMOGRAFO PROGRAMMABILE

Per chi voglia cimentarsi praticamente nelle applicazioni proponiamo un semplice esercizio: provate a modificare il codice di programma al fine di realizzare allarme temporizzato programmabile. Coloro che

Per approfondire

[1] PIC16F87X Data Sheet 28/40-Pin 8-Bit CMOS FLASH Microcontrollers (www.mikrochip.com)
[2] Help in linea MikroPascal Pro for PIC v. 5.61
[3] MikroPascal PRO for PIC v.5.61 demo e documentazione relativa (www.mikroe.com)
[4] Development Tools Easy PIC 3-Board Schematic with functional schematic;
[5] Orologio digitale con display a sette segmenti" -FE 307-Autore: A. Giannico

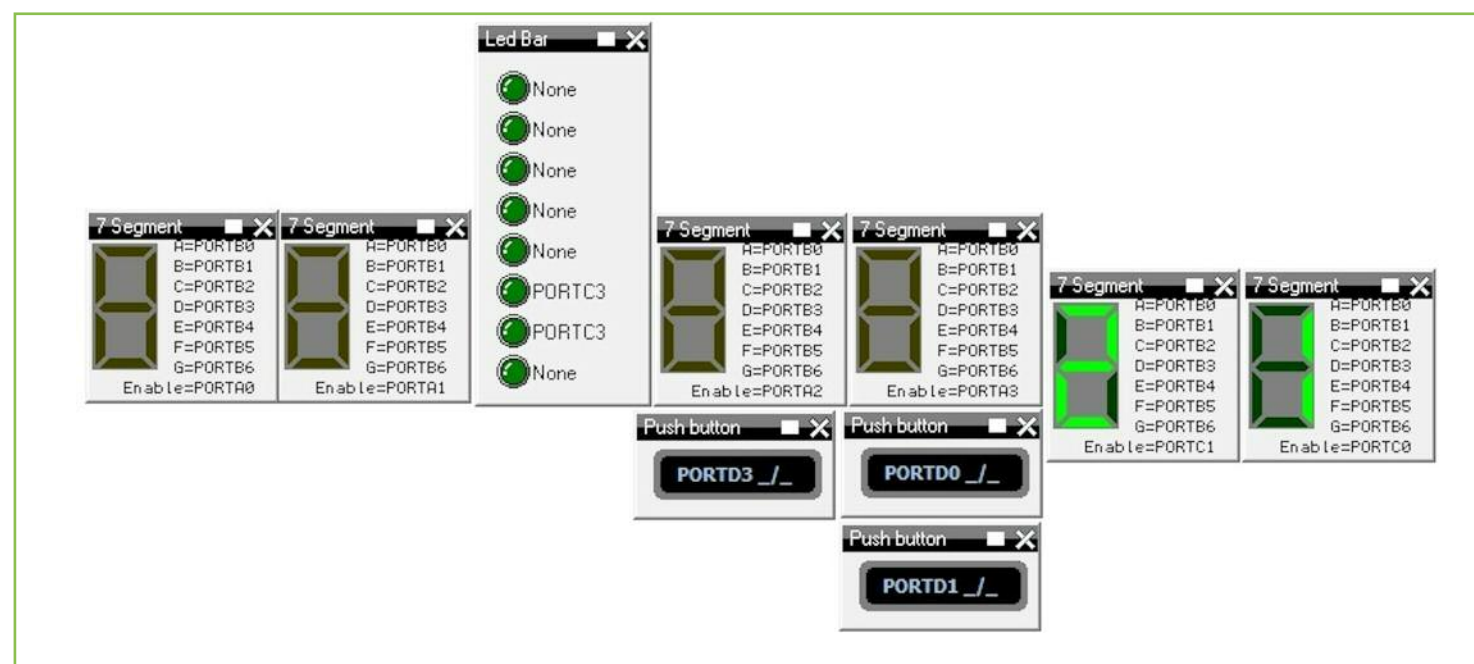



Figura 10-Simulazione dell'applicazione attraverso il simulatore Real PIC Simulator v.1.1.0.0 (stato StatoProg=1: programmazione del campo secondi prima che sia premuto uno dei pulsanti INCREMENTO e DECREMENTO che ne producono l'azzeramento)

Figura 13- Simulazione dell'applicazione attraverso l'interfaccia Real PIC Simulator v.1.1.0.0 (StatoProg=3: programmazione del campo ore; attraverso la pressione dei pulsanti INCREMENTO e/o DECREMENTO è possibile correggere il valore del campo "ore")

possiedono un bromografo artigianale potranno utilizzare questa applicazione per aggiungere al proprio bromografo la funzione di spegnimento automatico.
L'applicazione potrà in questo modo limitarsi al conteggio *hh:mm:ss* ma in maniera analoga è possibile realizzare un cronometro *hh:mm:ss:dd* che effettui il conteggio fino ai decimi di secondo e che integri le comuni funzioni di stop, ripartenza e azzeramento oppure un conto alla rovescia programmabile.

CONCLUSIONI

Al di là dell'applicazione mostrata in questa lezione, va sottolineato che l'impiego di display a sette segmenti, a causa della necessità del multiplexing, è in realtà spesso meno agevole e flessibile dell'impiego di LCD. Come avremo modo di mostrare in alcune applicazioni di esempio delle prossime lezioni e come abbiamo già avuto modo di intuire nel corso delle lezioni 5 e 6, l'impiego di un LCD risulta non solo più agevole ma anche più flessibile poiché consente di visualizzazione più informazioni, per esempio la temperatura rilevata da un sensore di temperatura e l'orario, su uno stesso visualizzatore. La cosa diviene ancora più agevole con l'impiego di display grafici. Per questo motivo, prima di dedicare alcune lezioni ad applicazioni di questo tipo, nella prossima puntata parleremo di display grafici controllati attraverso codice MikroPascal PRO for PIC.

Figura 11- Simulazione dell'applicazione attraverso il simulatore Real PIC Simulator v.1.1.0.0 (StatoProg=1: azzeramento del campo secondi attraverso la pressione di uno dei pulsanti INCREMENTO e DECREMENTO)

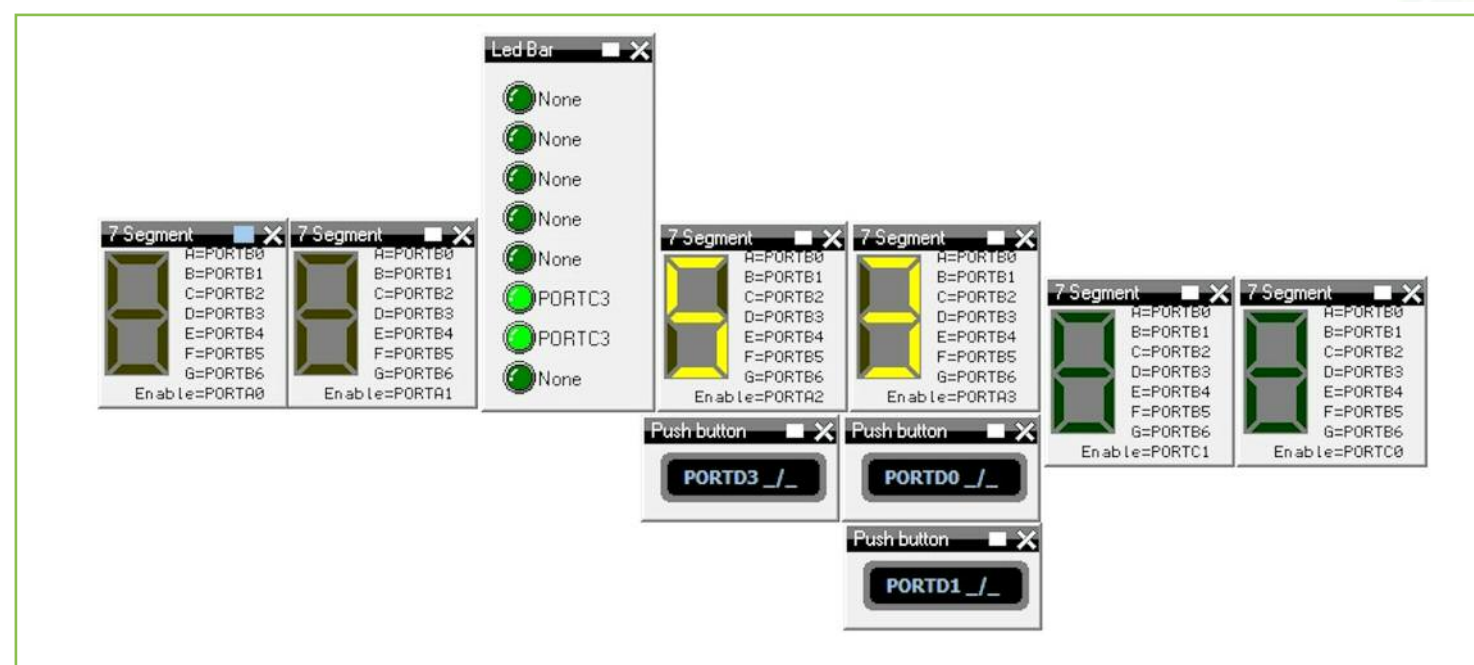



Figura 12- Simulazione dell'applicazione attraverso il simulatore Real PIC Simulator v.1.1.0.0 (Stato Prog=2: programmazione del campo minuti; attraverso la pressione dei pulsanti INCREMENTO e/o DECREMENTO è possibile correggere il valore del campo "minuti")

di MARCO MANGIAT

INTERVALLOMETRO FOTOGRAFICO

Da un appassionato di fotografia e di elettronica, ecco un utilissimo strumento per effettuare scatti fotografici ad intervalli di tempo prestabiliti. Il tutto con l'aiuto di Arduino

Una delle mie passioni è la fotografia e per realizzare filmati con la tecnica del "time-lapse" ho deciso di coniugare l'altra mia passione e costruire da me l'intervallometro da associare alle fotocamere che possiedo.

Navigando su internet ho scoperto Arduino e le sue potenzialità, quindi per prima cosa mi sono procurato un Arduino Uno.

Ho rinfrescato i ricordi relativi alla programmazione in C ed ho sviluppato il programma di controllo e l'hardware associato.

L'intervallometro è composto sostanzialmente da 4 parti:

- la scheda Arduino Uno
- un display LCD 16x2 righe
- un tastiera con 4 pulsanti
- i componenti da accoppiare ad Arduino per il controllo della fotocamera

Attualmente il cablaggio è realizzato su una bead-bord, in seguito il tutto verrà montato su un circuito stampato e collocato in un contenitore di adeguate dimensioni per essere facilmente trasportato.

Sul display vengono proposti i parametri che consentono di pianificare gli scatti (vedi figure).

Una volta confermati tutti i parametri ha inizio la ripresa e sullo schermo vengono proposti:

- Numero dello prossimo scatto che deve essere eseguito
- Secondi che mancano all'esecuzione dello scatto

La conclusione dei cicli è segnalata da un messaggio; premendo un tasto si torna alla definizione dei parametri.

TASTIERA

Iniziamo con la realizzazione della tastiera che è composta da quattro pulsanti disposti a croce:

I controlli associati prevedono:

Elenco componenti	
R1 R2 R3 R4	330 Ω
R5	4,7 KΩ
P1 P2 P3 P4	

- Prev - visualizza il parametro/menu precedente
- Next - visualizza il parametro/menu seguente
- \+ - incrementa il valore del parametro
- – - decrementa il valore del parametro

I tasti fanno parte di un circuito partitore di tensione che verrà letto dalla porta A0 di Arduino.

Per gestire in modo adeguato le voci proposte sul display il programma di controllo deve svolgere le seguenti attività:

- rilevare quale dei 4 tasti è stato premuto
- evitare il "rimbalzo" alla pressione del tasto
- attivare l'autoripetizione se si mantiene la pressione sul tasto; in tal modo si evita di continuare a cliccare.
- se su un tasto è mantenuta la pressione per un certo numero di ripetizioni, l'incremento / decremento del valore viene

aumentato a 10; al suo rilascio tale valore deve essere resettato.

Per quanto riguarda i primi tre punti ho pescato nel web che propone soluzioni in molteplici salse; per il quarto punto ho dovuto letteralmente inventare.

Figura 1: Attesa per attivazione autofocus (da 0 a 999 msec)

Figura 2: Numero dei cicli di scatto (da 1 a 999)

Figura 3: Tempo apertura otturatore (da 0 a 59m 59s)
NB: Per utilizzare i tempi di otturazione impostati sul intervallometro, è necessario attivare la modalità bulb sulla fotocamera.

Figura 4: Intervallo tra gli scatti (da 1 a 24h 59m 59s)
NB: Il tempo tra uno scatto e l'altro deve essere maggiore del tempo di apertura del otturatore, in caso contrario sarà il programma a modificare i parametri.

Figura 5: Tempo attesa antecedente al primo scatto (da 5 a 59m 59s)

Figura 6: Tempo complessiva del filmato in secondi con possibilità di variare i FPS per verificarne la durata (FPS 10 a 40)

Figura 7: Tempo totale della ripresa espresso in Giorni – Ore – Minuti. Da questo menù, premendo il tasto '+' o "-" inizia il ciclo di ripresa.

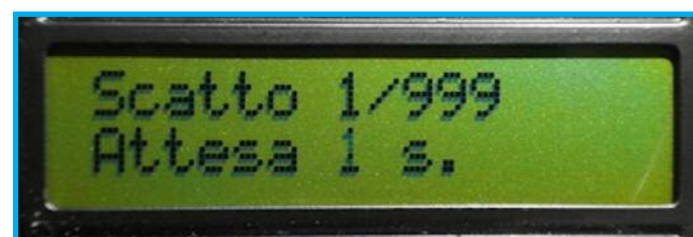



Figura 8.

Figura 9: Fase esecuzione sequenza di scatti

Figura 10: Fine del ciclo di scatti

INTERVALLOMETRO

Dopo aver analizzato il funzionamento del tastierino vediamo come funziona la parte software del intevallometro; nel listato 1 è riportata la logica del programma:

I cablaggi da realizzare non sono complessi, in figura 16 lo schema. L'alimentazione è attivata da I1 ed il led L1r segnala l'accensione, la retroilluminazione del display è azionata da I2 (ho messo R8 per attenuarne la luce), il trimmer P1 regola il contrasto (io ne ho usato uno che avevo in casa ma ne basta uno da 100Okm).

I due fotoaccoppiatori attivano l'autofocus e l'otturatore, i led L2v ed L3g ne evidenziano lo stato.

Vediamo come funziona lo scatto remoto

Tutte le fotocamere (bridge o reflex, escludendo le compatte), montano un connettore per il comando a distanza. Se possedete il comando remoto, apritelo e vedrete che sostanzialmente è null'altro che un doppio interruttore; il primo scatto attiva l'autofocus mentre il secondo l'otturatore.

La seconda fotocamera che voglio controllare è la Lumix FZ100 per la quale il cablaggio del terminale remoto è un poco particolare; nello stesso ingresso sulla fotocamera vengono gestiti il microfono esterno e lo scatto remoto.

Il connettore è uno spinotto da 2,5 mm a 4 poli ed utilizza la tecnica del partitore di tensione per controllare lo scatto.

Questo è per la Panasonic Lumiz FZ 100:

Figura 11 : Disposizione tasti

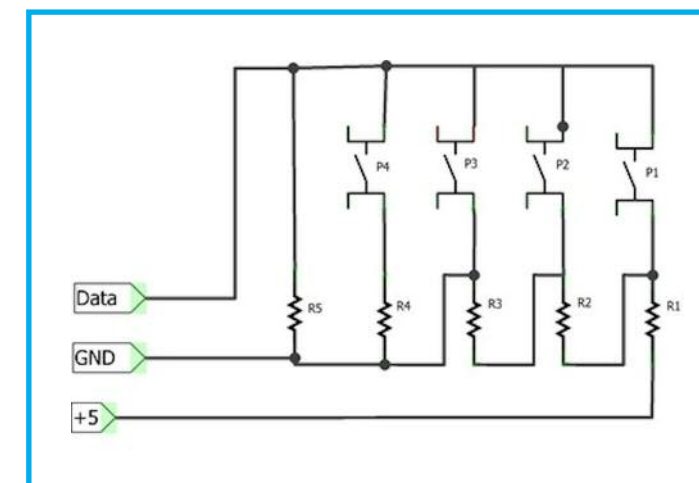



Figura 12 : Schema tastierino

Per realizzare la connessione alla fotocamera, potete eliminare la pulsantiera dello scatto remoto e utilizzare il cavo ed il connettore; per le fotocamere con partitore dovete ripristinare i cablaggi delle resistenze.

Nel mio caso, non volendo perdere la possibilità di utilizzare lo scatto remoto originale, interporrò dei connettori stereo da 3,5mm (uno M e uno F) per poter sostituire la pulsantiera originale con l'intervallometro.

In tal modo avrò la possibilità di controllarle entrambe le fotocamere sostituendo il cavetto di raccordo.

Quanto visto sopra, relativamente alla Lumix, vi fa capire la necessità di disaccoppiare i circuiti in uscita mediante fotoaccoppiatori; nel caso non lo avessi utilizzati avrei rischiato di danneggiare sia Arduino che la FZ 100.

PER CONCLUDERE

Ecco alcune immagini che filtrano dell'intrico dei cavetti; non spaventatevi è solo perché sono poco ordinato ..... e poi è la prima volta che uso una breadbord e Arduino.

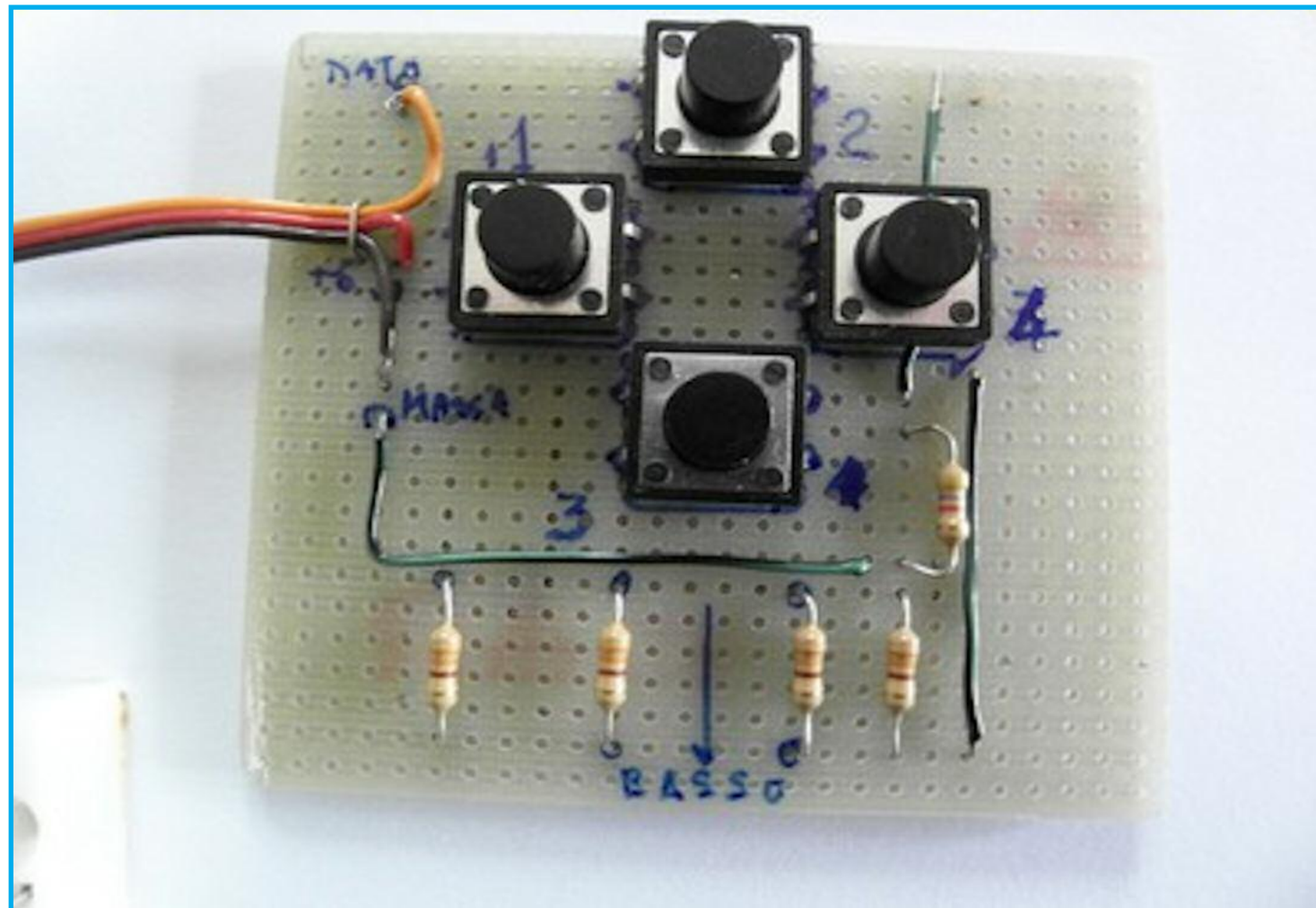



Figura 13: Il Tastierino

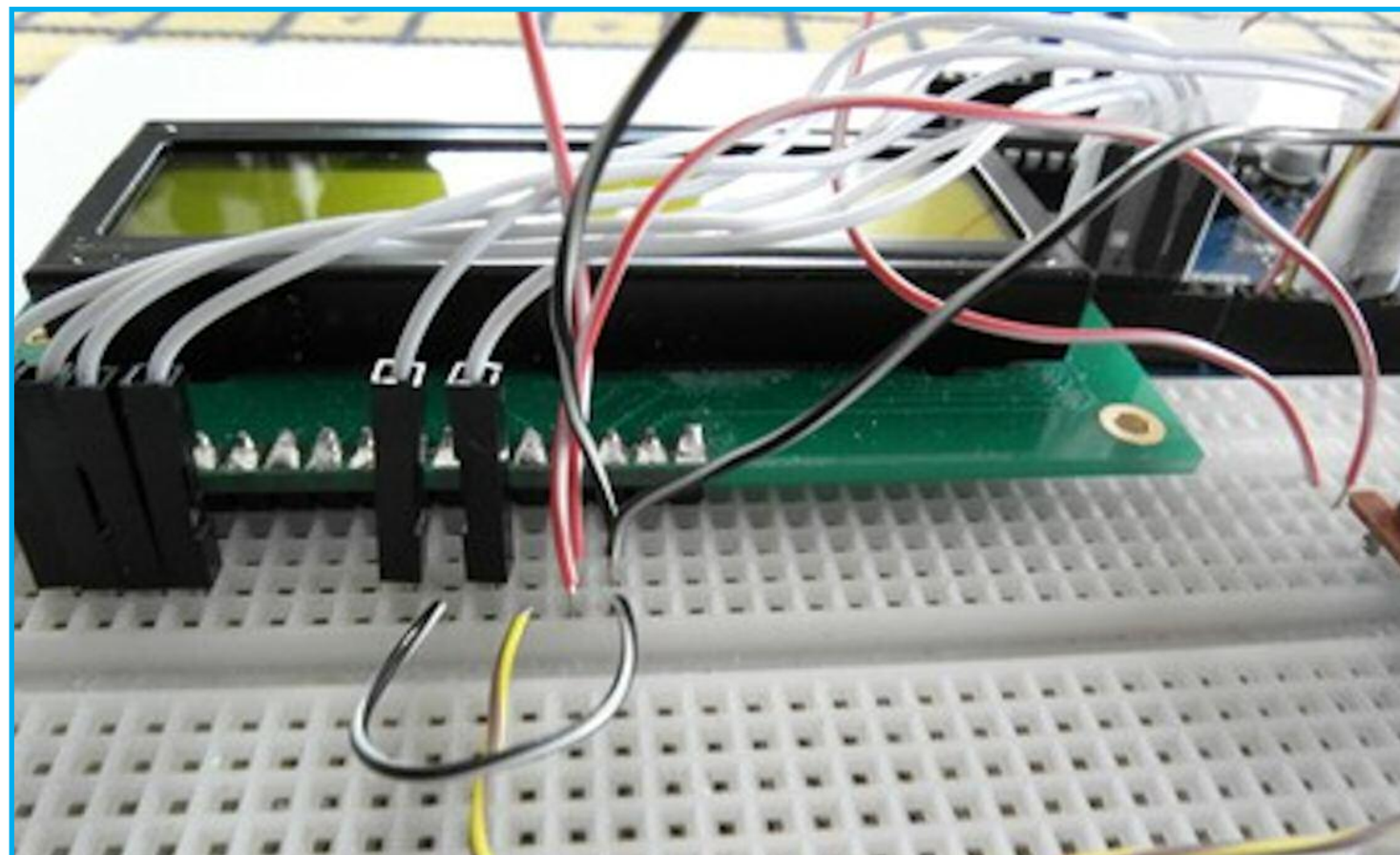



Figura 14: Connessioni al display

LISTATO 1

Download

```
caricamento della libreria per la gestione del display LCD
dichiarazione variabili
dichiarazione di alcune funzioni richiamate dall'interno del loop
setup iniziale
loop()
  se startIntervallometro == false allora definisco i parametri
           leggo la tastiera
           alla pressione tasto DESTRA attiva il menù/variabile successivo
           alla pressione del tasto SINISTRA attiva il menù precedente
           alla pressione del tasto SU incrementa il valore della variabile
           alla pressione del tasto GIU decrementa il valore della variabile
  se startIntervallometro == true allora inizio il ciclo dell'intervallometro
           se è il primo ciclo, attendo i secondi impostati nella variabile
           varRitardoStart
     attivo autofocus
     attesa tempo autofocus

     attivo otturatore
     attesa chiusura otturatore
           se durante l'attesa premo per 1 sec. un tasto, i cicli vengono abortiti
           viene riproposta la definizione dei parametri
     chiude otturatore
     disattivo autofocus
     incremento il contascatti
     se sono stati eseguiti tutti gli scatti previsti viene riproposta la
     definizione dei parametri
     attesa, per i secondi definiti nella vaiabile varAttesa, prima di iniziare
     il ciclo seguente
     se durante l'attesa premo per 1 sec.  un tasto, i cicli vengono abortiti
     e viene riproposta la definizione dei parametri
fine loop
```

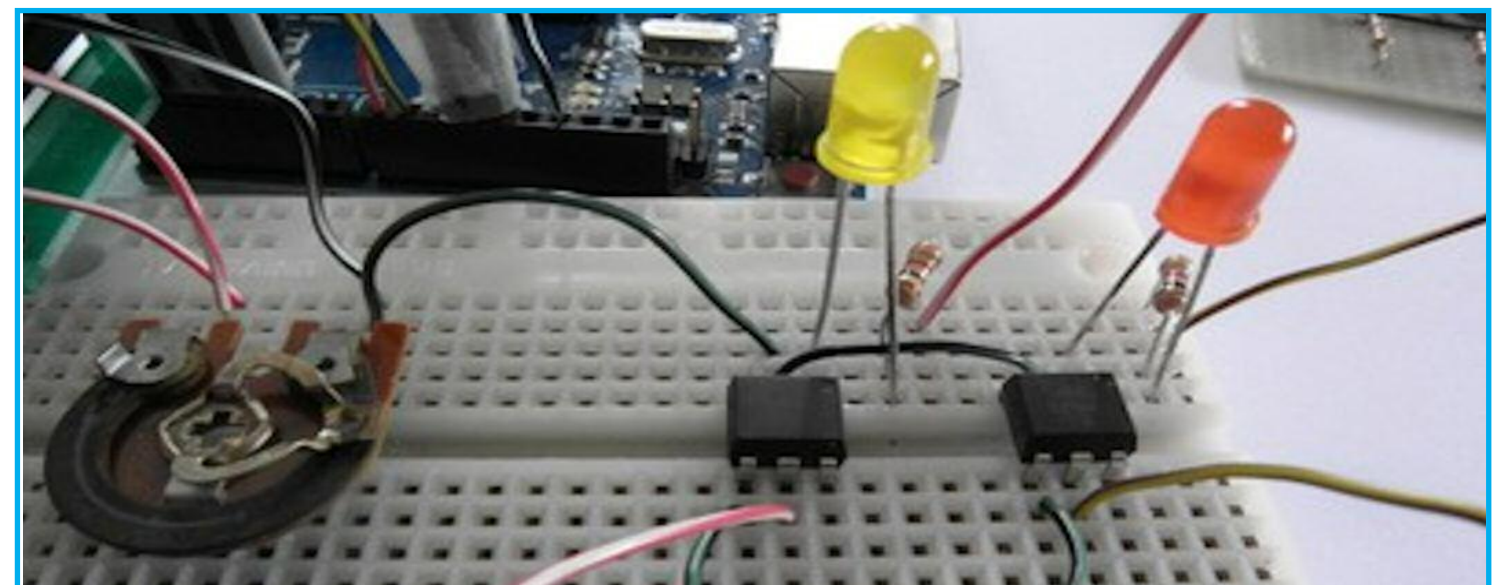



Figura 15: Regolazione contrasto display e fotoaccoppiatori

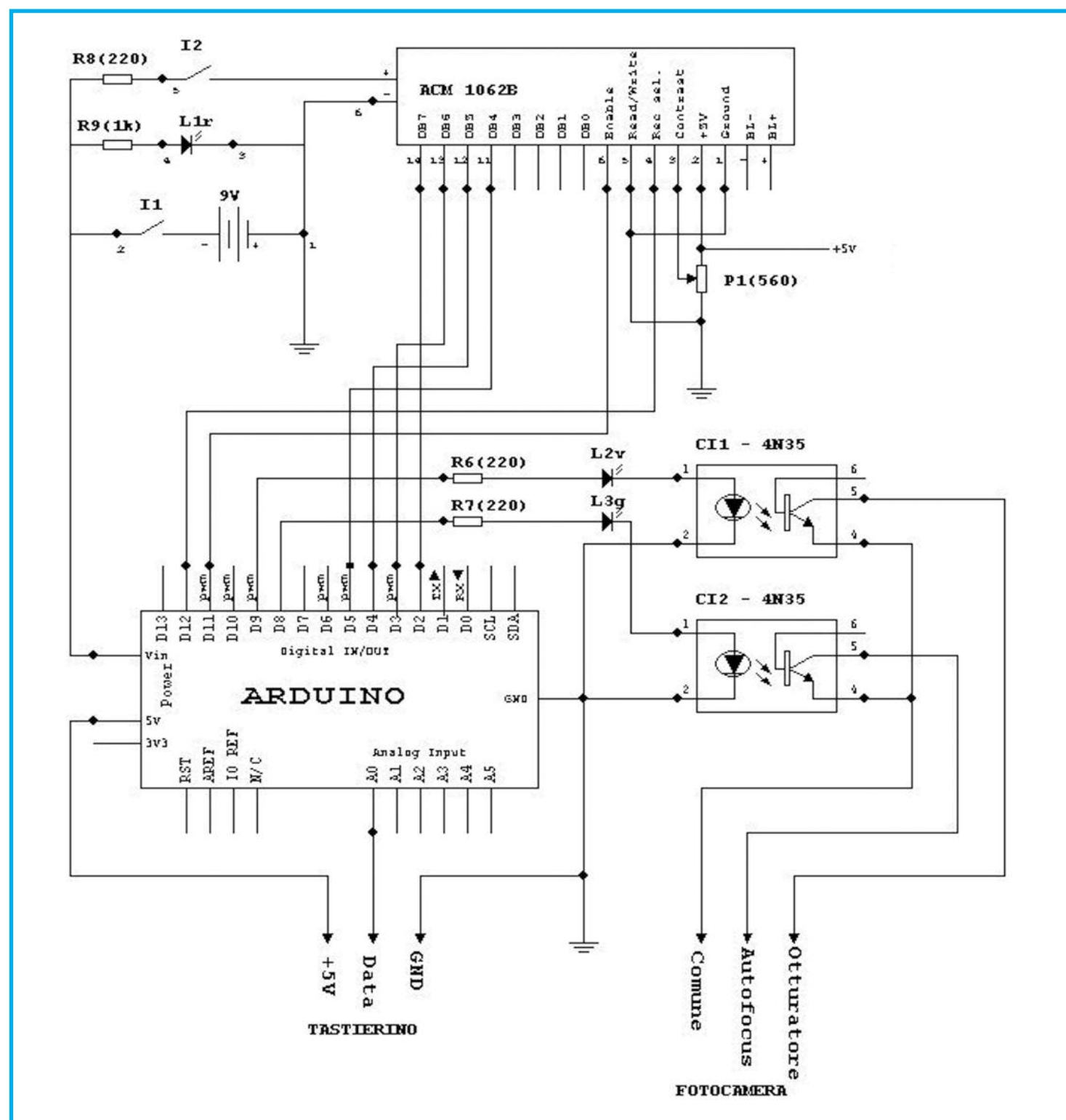



Figura 16: Schema intervallometro

Visto che su Arduino sono disponibili ancora alcune connessioni digitali, pwm ed analogiche, è possibile espandere il progetto inserendo controlli per trappole fotografiche, luci, flash e . . . . . la fantasia non ha limiti!

Buon cablaggio e buone fotografie a tutti.

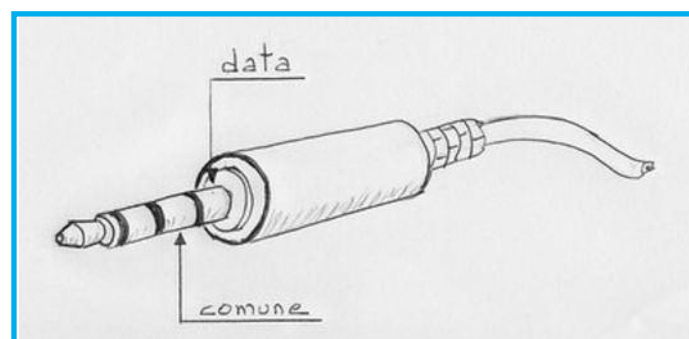



Figura 17: Connettore FZ100

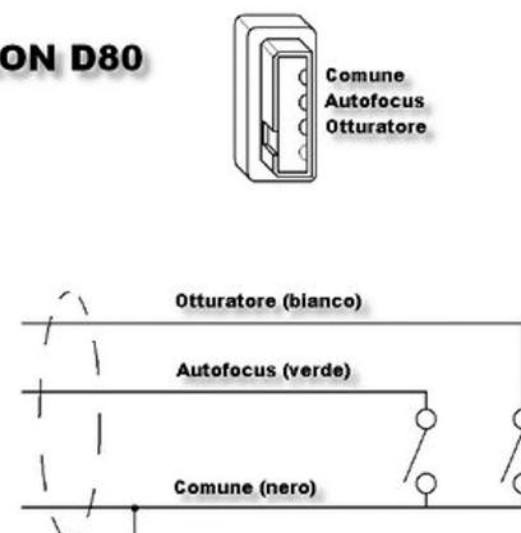



Figura 18: Contatti scatto remoto Nikon

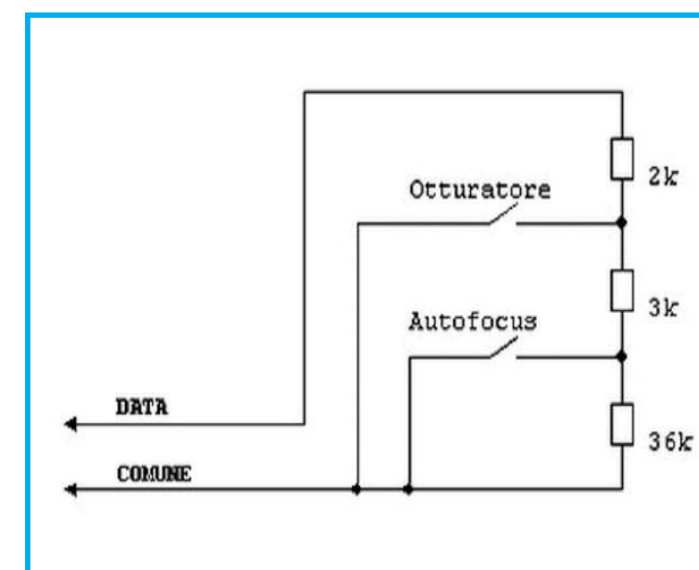



Figura 19: Contatti scatto remoto FZ100

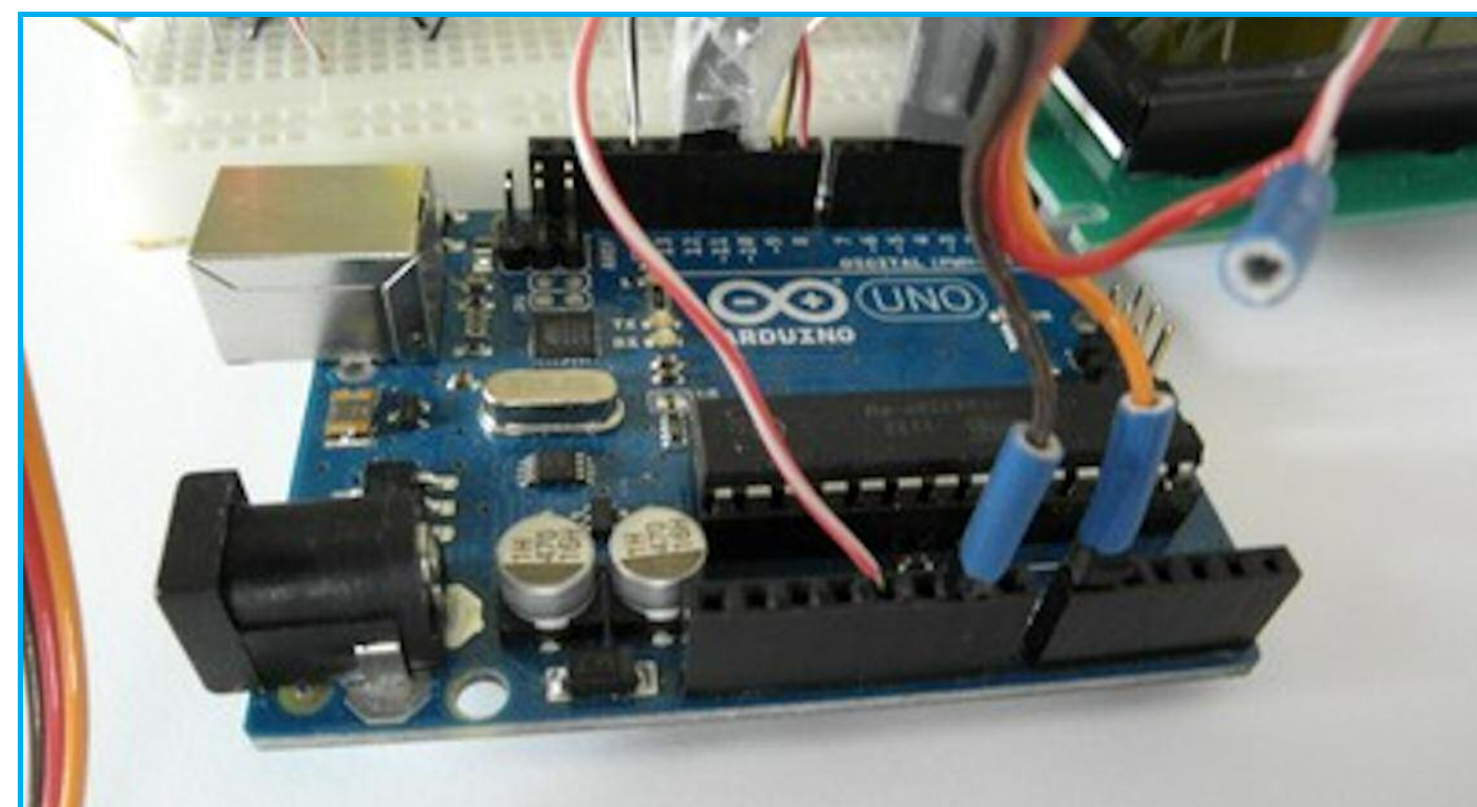



Figura 20: Arduino Uno

Elenco componenti	
	Arduino Uno
	display AMC 1062B (16colonne x 2 righe)
	fotoaccoppiatori 4N35
I1 - I2	interruttori
L1r	led rosso (accensione intervallometro)
L2v	led verde (attivazione otturatore)
L3g	led giallo (attivazione autofocus)
	pila 9V
P1	trimmer 560
R9	resistore 1k
R6 – R7 – R8	resistori 220

di MARIO MOTTULA

SISTEMA DI CONTROLLO DI UN MOTORE CC

Il seguente articolo ha lo scopo di spiegare come possono essere controllati i motori in corrente continua di una certa potenza, con la possibilità di inversione del verso di rotazione, con la tecnica di controllo a ponte H. Vedremo dunque, come sarà possibile mediante la modulazione a larghezza di impulsi PWM, sia un controllo con incremento/decremento della velocità, che l'inversione della marcia

Il progetto è stato sviluppato per il controllo di un motore di un modellino di jeep radiocomandata, prelevando come segnale di comando quello destinato ad un servo che non veniva utilizzato.

Vedremo comunque come rendere generalizzabile il progetto in modo da poterlo utilizzare anche non avendo a disposizione il segnale di comando di un servo e spiegando per sommi capi, come sarà possibile comandare un motore in DC partendo ad esempio da due pulsanti.

IL CIRCUITO ELETTRICO

Partiamo dallo stadio di alimentazione formato dall'integrato stabilizzatore uA7805 e dai due condensatori C11 e C12, come visibile in figura 1.

All'ingresso del regolatore di tensione, mediante il connettore P1, arriverà la tensione stabilizzata proveniente da una sorgente esterna oppure, come nel mio caso da una batteria agli ioni di litio.

La tensione massima accettabile sugli ingressi VCC dei driver dei mos IR2110 (U4 ed U5), è pari a 20 Volt come da datasheet, nel nostro caso abbiamo un ingresso di 14 Vdc.

Dunque, dando un'occhiata allo schema, si capisce che la nostra sorgente di alimentazione esterna, che collegheremo al connettore P1 dovrà innanzitutto non essere mai maggiore a 20 Volt, ma nello stesso tempo dovrà essere scelta in modo da poter alimentare in modo corretto il motore che poi collegheremo all'uscita del ponte H.

Ad esempio, se sceglieremo un motore DC di 12 Volt, allora converrà alimentare il sistema con una tensione di poco superiore, ad esempio 13 – 14 Volt, inquanto come vedremo in seguito, sugli N-MOS , ci sarà una caduta di tensione tra Drain e Source dei due Mos Vds_{on}, che causerà un calo di tensione sul carico di circa 1 volt per ogni N-MOS.

Gli stessi 14 Volt vanno ad alimentare, sia i drain dei MOS Q1 e Q3 e sia il regolatore di tensione U6, che fissa la sua uscita a 5 Vdc i quali alimenteranno tutta la logica del nostro sistema.

La logica del sistema, è implementata dal firmware del PIC, il quale gestisce a seconda del segnale proveniente dalla scheda radio (fig 2), lo stato di avanzamento, blocco o inversione del senso del motore.

Nel paragrafo descrizione di un segnale di comando si un sevo, descriverò brevemente cos'è questo componente e com'è composto un segnale che permette il suo controllo.

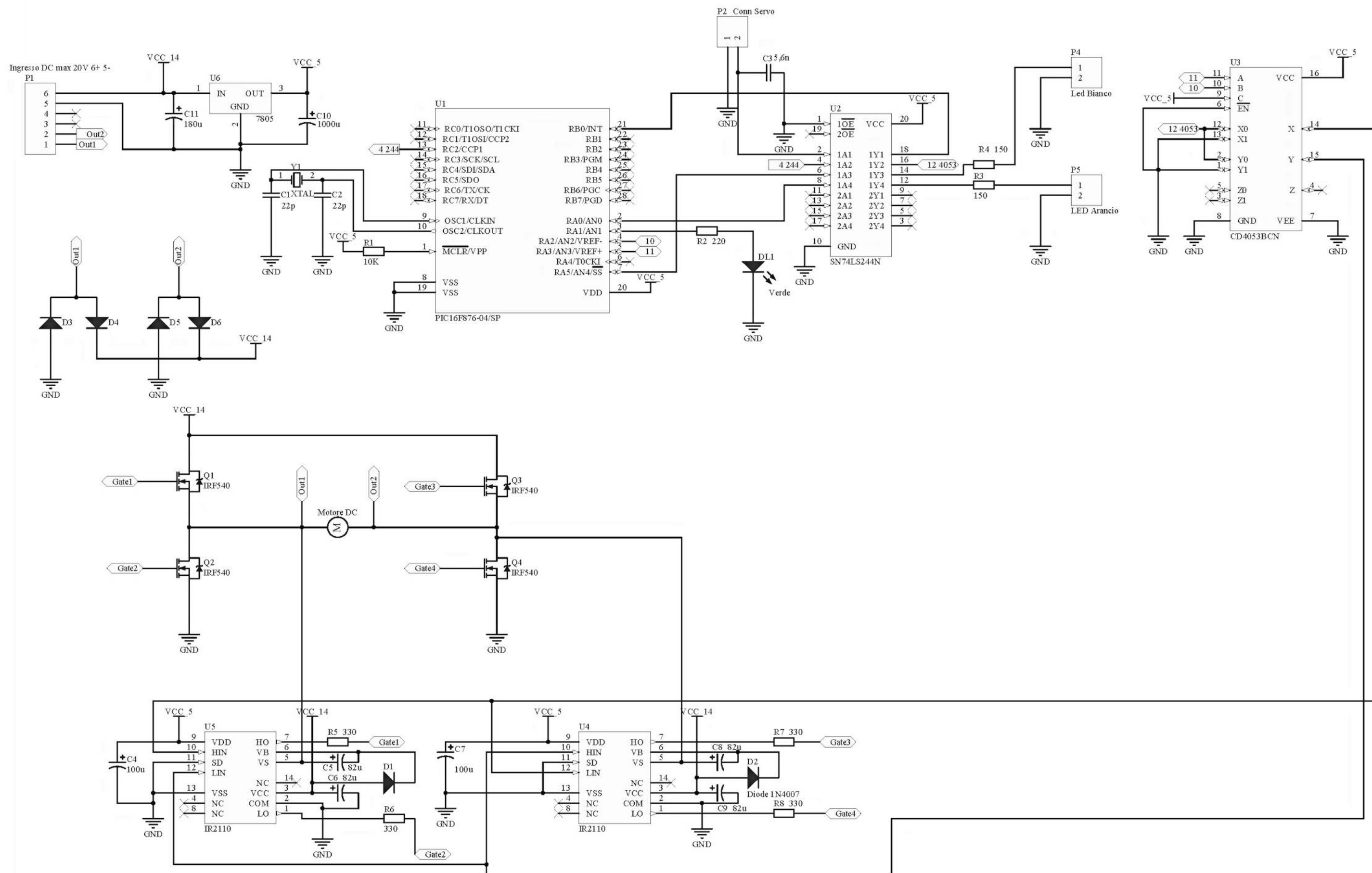
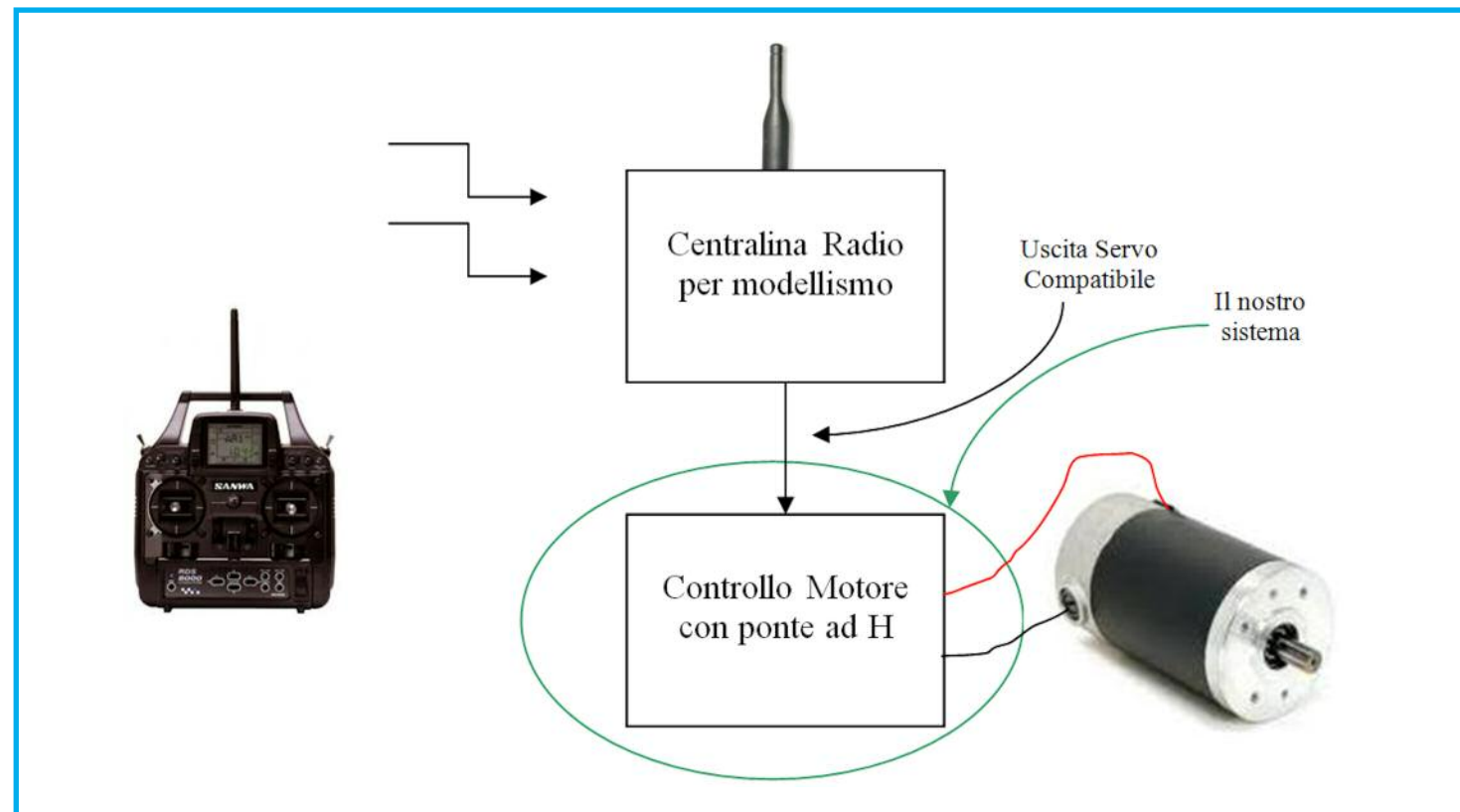



Figura 2

Andando avanti con la descrizione dello schema elettrico, ci soffermiamo per un momento sul connettore P2.
Come visibile questo connettore è formato da due pin, ma in realtà sulle schede radio che hanno un uscita di comando idonea a pilotare un servo, i pin sono 3 ovvero, uno massa uno di alimentazione ed uno di comando.
A noi servirà quindi inviare sul pin 1 di P1 la massa, mentre sul pin 2 di P2 un segnale che ha un ampiezza pari a 5 Volt ed una durata che va da 1 a 2 ms (vedi paragrafo Descrizione del segnale di comando di un servo).
Se abbiamo a disposizione quindi una ricevente radio, sarà sufficiente prelevare dall'uscita che comanda un servo, i segnali di massa e di comando.
Il segnale di comando quindi, dal pin 2 di P2 andrà in un 74ls244 (U2) che provvederà innanzitutto a bufferizzare gli ingressi e le uscite del PIC ed inoltre a rendere perfettamente compatibile il segnale proveniente da un'eventuale scheda radio, con l'ingresso RB0 del micro.
Oltre ad U2 vediamo l'integrato U3 che è un multiplexer CD4053, il quale opportunamente comandato dal PIC farà in modo di convogliare il segnale PWM generato dal pin RC2 del PIC , che transita anch'esso attraverso il buffer, in uno o nell'altro ramo del ponte ad H, a seconda che si voglia comandare il motore in un verso o nell'altro.
In concomitanza di un segnale di comando inviato dal PIC del tipo RA2 = 0 , RA3=1, allora il multiplexer porterà sull'uscita X (pin 14 di U3) il segnale PWM applicato sul sui suoi ingressi X0,Y0.
Nel caso contrario, ovvero che il PIC invia al mux i segnali RA2=1 ed RA3=0, allora il segnale PWM verrà deviato sull'uscita Y.
In questo modo potranno essere alimentati alternativamente i due rami del ponte, dando dunque il verso di rotazione del motore, mentre invece la velocità di rotazione verrà data dal duty-cycle del segnale PWM generato dal PIC.
Andando avanti nella descrizione dello schema elettrico, si possono notare i due integrati identici IR2110 siglati con U4 ed U5 .
Questi sono dei driver per MOS e IGBT e servono per l'appunto a comandare proprio nelle applicazioni di potenza questi componenti, limitando al minimo i fenomeni induttivi che scaturiscono dalle frequenti commutazioni sui gate e fornendo i valori adeguati di tensione Vg e Vs i quali fanno si di ottenere le condizioni di saturazione dei MOS.
Ottenere la saturazione in un N-MOS non è banale con una logica alimentata a 5 Volt, e con carichi da pilotare dell'ordine di qualche decina di Volt.
Infatti se volessimo erogare la massima corrente di drain verso il carico, in una configurazione circuitale come la seguente, dovremmo far lavorare il mos in saturazione, soddisfacendo le seguenti condizioni:

1) Vgs>Vt
2) Vgs>Vds+Vt

Ricordando che la condizione di saturazione di un N-MOS è la seguente:

Vds=Vgs-vt

Detto ciò, realizzare la condizione 2) in un circuito che pilota il carico con una tensione di 12 Volt, e la sua logica di comando è alimentata a 5 non è cosa semplice, infatti in una condizione del genere, per avere sul carico una tensione di 12 Volt circa a corrente pressoché costante, dovremmo applicare sul gate una tensione di 16Vdc, vedi figura 3 (Caso2).

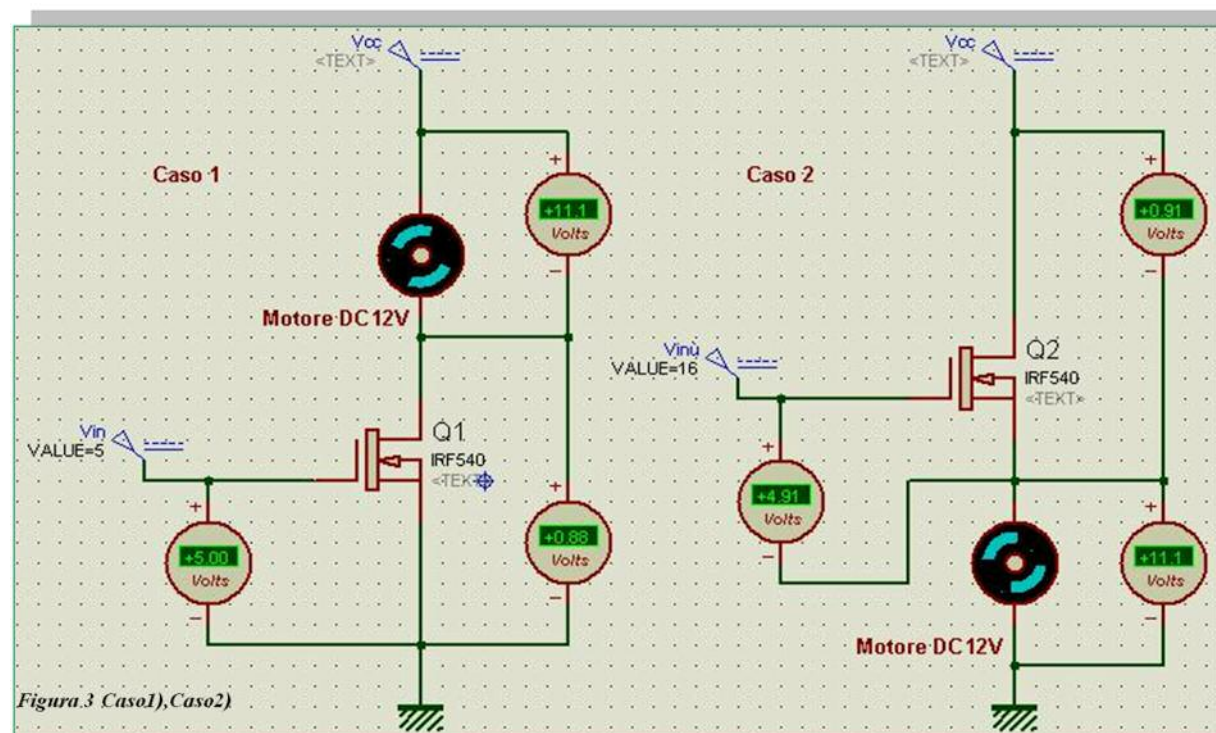



Figura 3

Recommended Operating Conditions

Symbol	Definition		Min.	Max.	Units
V_B	High side floating supply absolute voltage		V_S + 10	V_S + 20	V
V_S	High side floating supply offset voltage	(IR2110)	Note 1	500	
		(IR2113)	Note 1	600	
V_{HO}	High side floating output voltage		V_S	V_B	
V_{CC}	Low side fixed supply voltage		10	20	
V_{LO}	Low side output voltage		0	V_{CC}	
V_{DD}	Logic supply voltage		V_{SS} + 3	V_{SS} + 20	
V_{SS}	Logic supply offset voltage		-5 (Note 2)	5	
V_{IN}	Logic input voltage (HIN, LIN & SD)		V_{SS}	V_{DD}	
T_A	Ambient temperature		-40	125	°C

Typical Connection

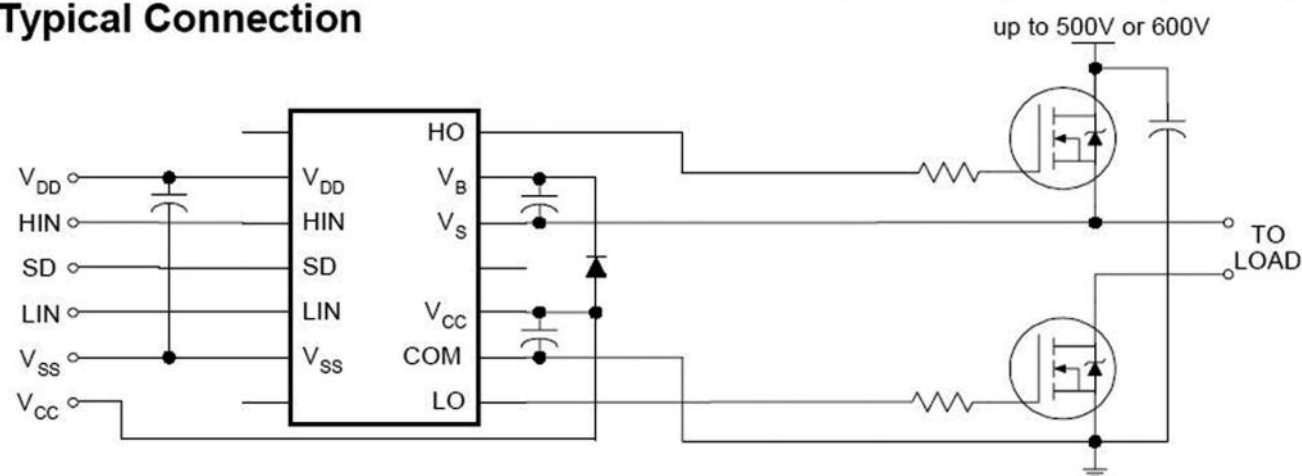
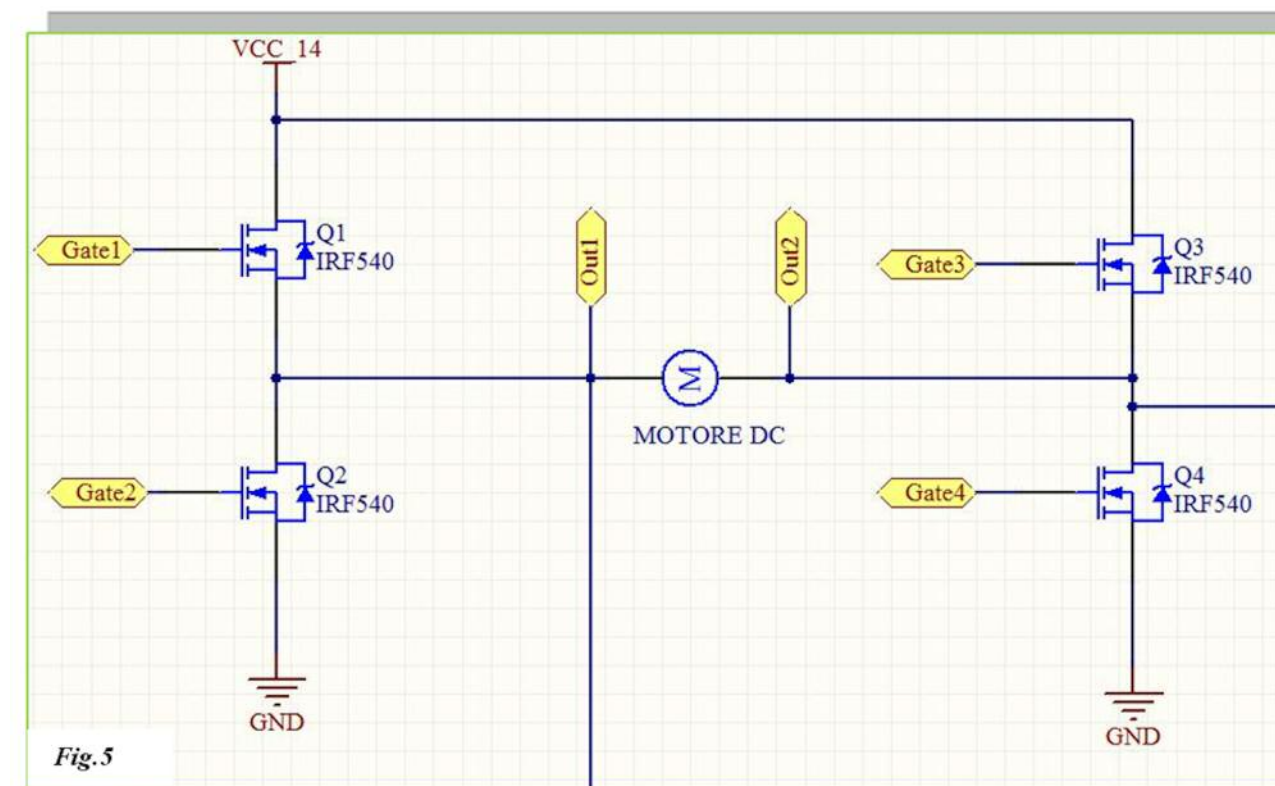



Figura 4

Figura 5

La situazione cambia invece nello schema sempre di figura 3 ma (Caso 1), in cui si nota come per soddisfare la condizione **Vgs>Vds+Vt**, basta un ingresso di 5 Volt, in quanto in tal caso il source essendo collegato a GND ha un potenziale nullo e quindi non sottrae nulla alla Vgs che è pari alla Vg.La Vds a sua volta è limitata dalla caduta di tensione sul motore, e quindi come vediamo in figura è di circa 900 mV, il che facilita l'ingresso in saturazione del MOS.
Di conseguenza in un circuito come quello di figura 3 (Caso1) il controllo è comodo, ma se siamo nel caso 2 allora servono tecniche di comando più articolate.

Nel nostro caso con un ponte ad H, ci troveremo ad affrontare entrambe le problematiche precedenti, che verranno risolte alternativamente dai driver IR2110.
Quindi un altro compito importante di questo dispositivo, come già accennato precedentemente, sarà quello di pilotare i gate ed i source dei MOS con livelli di tensione idonei al caso in esame.
Osservando il datasheet del componente, si nota come il dispositivo è idoneo ad arrivare a tensioni di controllo sul Source dei MOS elevatissime, eliminando qualunque problema di conducibilità.
Nella figura 4 vi è una configurazione tipica dell'IR2110 con la tabella che riassume i valori di tensione applicabili e prelevabili dagli ingressi e sulle uscite.

PONTE AD H

In questo paragrafo descriverò il funzionamento della parte di potenza del nostro sistema, ovvero il ponte ad H o H-Bridge.
Quest'ultimo rappresenta una configurazione circuitale, come nel nostro caso, in cui il motore viene pilotato da quattro MOS o BJT, che fungono da interruttori (fig.5).
Gli interruttori, che formano un lettera H dal loro posizionamento, nel nostro caso sono Q1,Q2,Q3 e Q4, assieme al motore in corrente continua. I due mos Q1 e Q3, vengono detti di source o high-side switch, mentre i mos Q2 e Q4 vengono detti low-side switch o di sink. E' un dispositivo che funziona a quattro quadranti nel piano (I,V), ovvero a fronte di una alimentazione non duale, il carico può essere sottoposto a tensione negativa o positiva. Il metodo di funzionamento scelto è tipicamente chiamato ***Sign-Magnitude PWM***, in cui il segnale PWM viene inviato contemporaneamente o ai mos Q1 e Q4, oppure ai mos Q3 e Q2. L'invio del segnale PWM alle due coppie di mos, viene effettuato dal multiplexer comandato opportunamente dal PIC. Come vedremo dal firmware infatti, il pic in base al segnale proveniente dal radiocomando, abiliterà il PWM o sugli ingressi Lin di U4 e Hin di U5, oppure sugli ingressi Hin di U4 e Lin di U5. I mos verranno dunque comandati dalle uscite di U4 ed U5 nominate Ho ed Lo (vedi schema elettrico di Fig.1). In questo modo il segnale PWM comanderà mediante il suo duty-cycle, la velocità in un verso o nell'al-

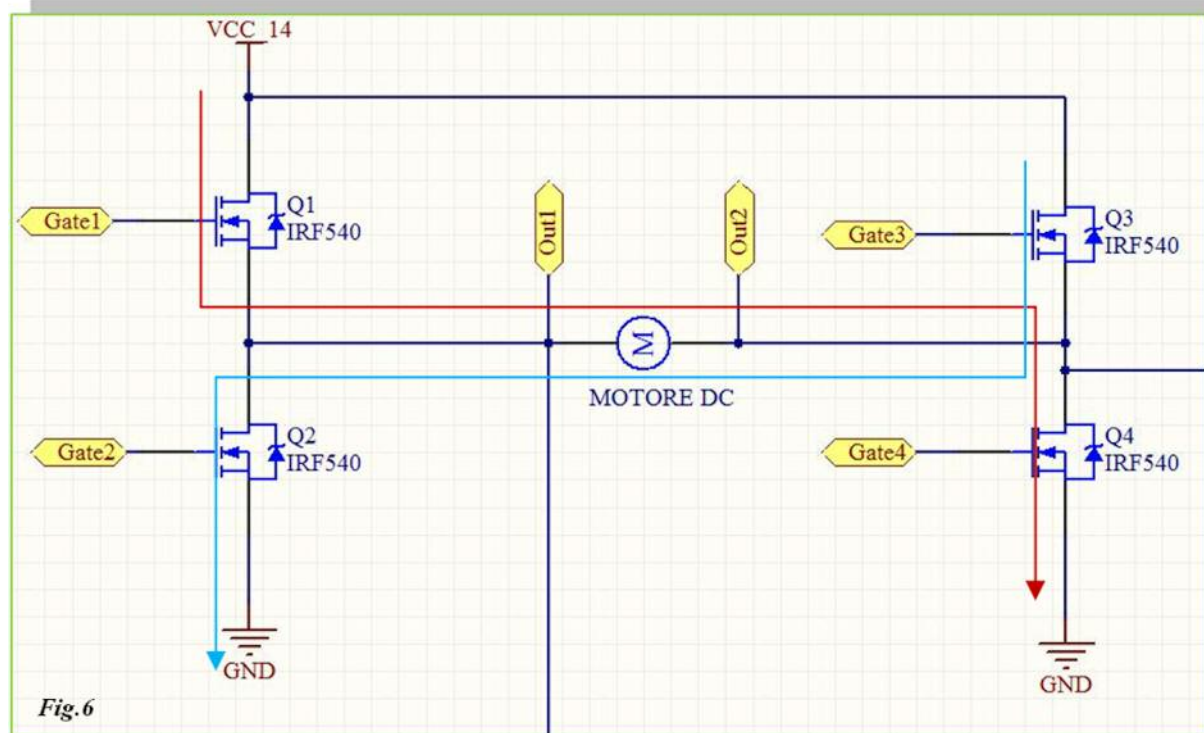



Figura 6

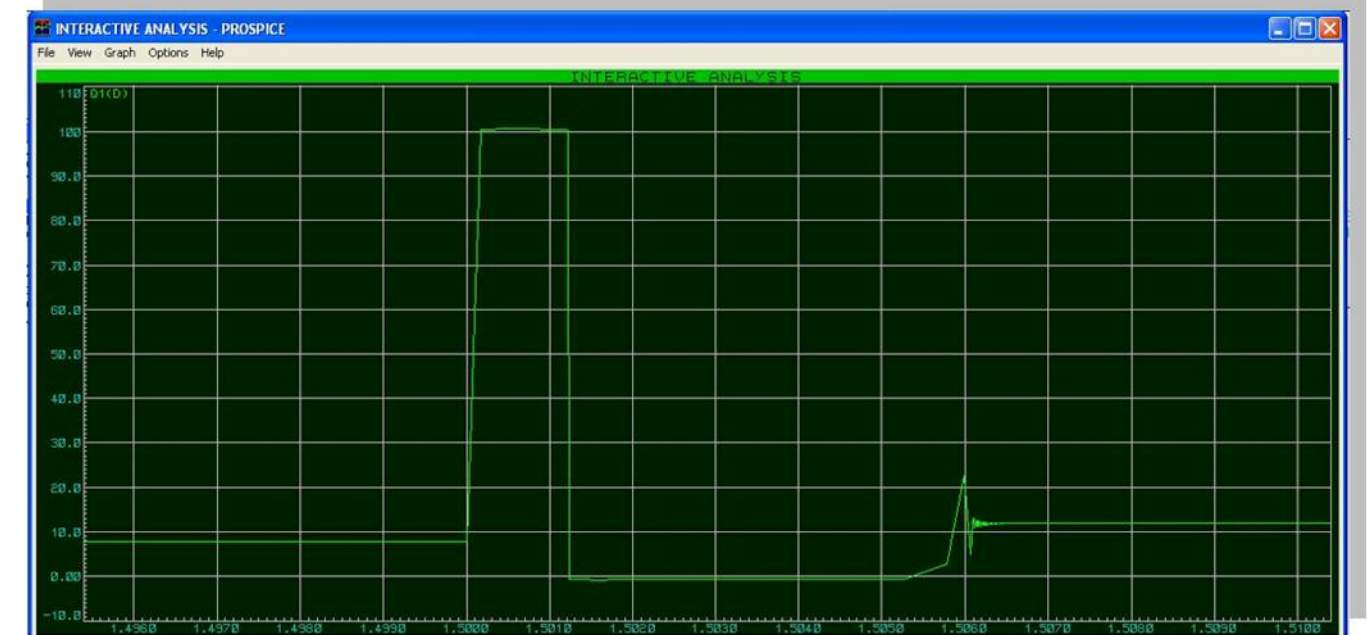



Figura 7

Figura 8

tro del motore. Il duty-cycle come vedremo, varierà dal suo valore minimo al suo valore massimo in entrambe le direzioni di rotazione, le quali saranno stabilite dall'opportuna accensione dei mos vedi fig.6. Il percorso rosso, evidenzia l'accensione dei mos Q1 e Q4, mentre quello azzurro l'accensione di Q2 e Q3. Le configurazioni di minimo e massimo corrisponderanno rispettivamente a:

- Duty-cycle = 0 %: motore fermo
- Duty-cycle = 50%: motore a media velocità
- Duty-cycle = 100%: motore a massima velocità

E' interessante notare come nello schema elettrico sono stati interconnessi tra Vcc_14 motore e massa, due coppie di diodi. Questi sono i cosiddetti diodi di ricircolo o di free Wheeling, che permettono la libera circolazione della corrente accumulata nel motore, quando viene interrotta la tensione ai suoi capi. Il motore in DC può essere schematicamente rappresentato come un induttore con in serie una resistenza interna, se questo viene attraversato da corrente allora la tensione ai suoi capi è regolata dalla relazione:

$$V=-L\ di/dt$$

In breve accade che: se bruscamente viene aperto uno dei MOS che comanda il motore, allora l'induttore tenterà di scaricare istantaneamente la corrente accumulata attraverso la resistenza interna e quella dei mos di comando. Scaricare la corrente in maniera istantanea, fisicamente significa avere una derivata temporale molto piccola, ma se ciò è vero allora dt tende a zero e di conseguenza V assumerà un valore elevatissimo. Se la tensione ai capi dell'induttore aumenta, ciò andrà a discapito del dispositivo di comando, ovvero del MOS che potrà subire danni permanenti.

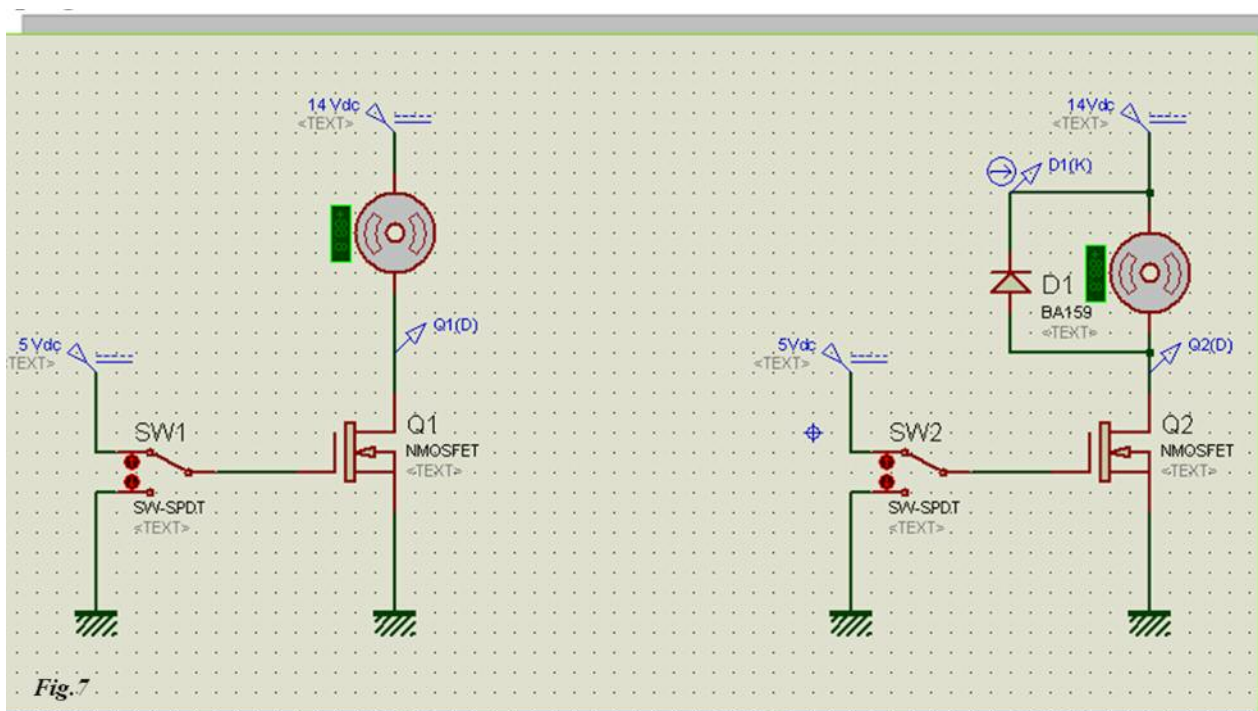
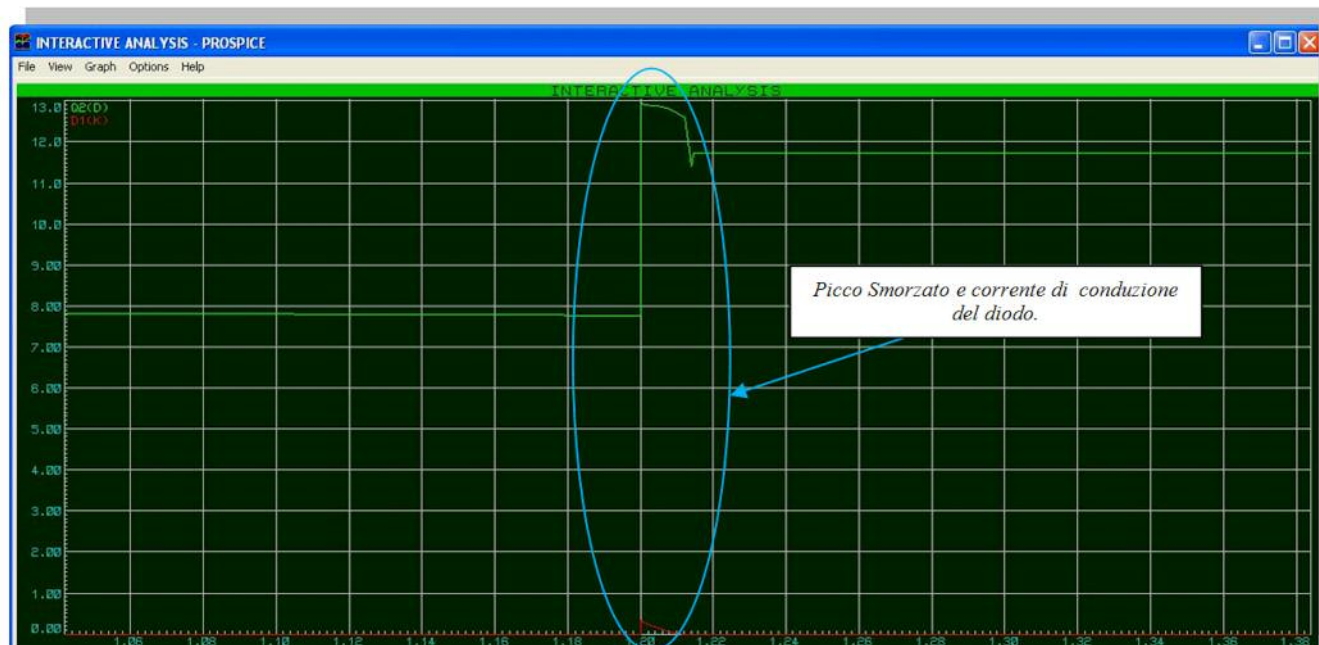



Figura 9

Se invece in antiparallelo all'induttore vi è un diodo di ricircolo, allora quando la tensione ai capi dell'induttore aumenta, questa farà andare in conduzione il diodo che farà circolare la corrente accumulata dall'induttore sull'induttore stesso, senza andare ad intaccare altre parti del sistema.
In figura 7 due esempi di comando di un motore in DC mediante N-MOS, che evidenziano quanto spiegato fino ad ora.
Nel primo caso consideriamo lo schema di sinistra, ovvero quello senza dio do di libera circolazione.
Durante la commutazione del MOS dallo stato ON a quello OFF, ci aspettiamo quanto descritto precedentemente, ovvero la tensione sul nodo Q1(d) cresce vertiginosamente (Fig.8).
Si può infatti vedere dalla simulazione, come in commutazione da on ad off, si ha uno spike di circa 100Volt su questo nodo.
In figura 9 invece si vede come l'inserimento di un diodo di ricircolo, abbassa significativamente il picco di tensione, portandolo a livelli normali.
Si può vedere anche l'entrata in funzione del diodo D1, il quale nel momento in cui la commutazione del MOS passa da ON ad OFF, entra in conduzione (vedi grafico rosso che rappresenta la corrente di libera circolazione D1(k) al momento del picco).
Di seguito dunque le simulazioni dei due casi precedenti.

IL FIRMWARE

Giunti alla fine dell'articolo, descriverò il firmware che gestisce la logica del sistema, proponendo un alternativa al comando, che così potrà non essere necessariamente il segnale proveniente dalla scheda di comando di un serve, ma magari da due pulsanti avanti/indietro.
Quest'ultimo è stato sviluppato mediante la piattaforma MikroC PRO.
Come si può notare prima del main, vengono definite tutte le variabili di sistema che si riassumono nei flag , nella variabile time che tiene conto della larghezza dell'impulso e della variabile contatore i necessaria nei cicli for.
Nel listato 1 si possono comprendere i modi di configurazione delle porte del PIC e dei registri interni. La parte fondamentale del firmware, viene gestita in un ciclo infinito (while(1)) nel quale si va a controllare il valore della larghezza dell'impulso in ingresso con dei valori empirici reimpostati.
In base dunque ala larghezza dell'impulso, si associa un valore alla funzione PWM che determinerà la velocità di rotazione del motore.
In fig.11 si nota come a seguito di un interrupt sulla porta rb0, la variabile time assumerà un certo valore rappresentativo della larghezza dell'impulso.
Se tale valore sarà compreso entro un certo intervallo, allora si inserisce il corretto verso di marcia ,con il settaggio della porta a2 ed a3 e poi si associa alla funzione PWM un determinato valore.
Da notare che la funzione PWM accetta valori che partono da 0 (minimo) a 255(massimo).
Nel progetto in esame non ho mai utilizzato il valore 255 per la funzione PWM, inquanto a livello pratico questo ha dimostrato vari malfunzionamenti, consiglio dunque di non superare mai il valore di 250.
Dal firmware allegato, si nota la ricorsività di questi if all'interno del while i quali svolgono la funzione descritta pocanzi.
Passiamo ora alla routine di interrupt, la quale come si può vedere dalla figura seguente, provvede a misurare la larghezza dell'impulso.
Vediamo in che modo: al momento dell'arrivo di un fronte alto sul pin rb0, allora il pic esce dal main e serve l'interrupt in esame, a questo punto il contatore interno del mi-

```
void main() {

        TRISB=0b00000001;       //Configurazione porta b come uscite tranne B0 che è ingresso si interrupt
        TRISA=0x00;             //Porta A configurata interamente come uscita
        TRISC=0x00;             //Porta C configurata interamente come uscita
        PORTA=0;
        PORTB=0;
        adcon0=0x00;            //Disabilito convertitore ADC
        adcon1=0b00000110;      //Imposto porta A come I/O digitali
        INTCON.GIE = 1;         //Abilito gli interrupt non mascherabili
        INTCON.RBIF=0;          //Azzero il flag degli interrupt periferici
        INTCON.PEIE = 0;        //Disabilito gli interrupt periferici sulla porta b
        intcon.inte=1;          //Abilito 1 solo interrupt sulla porta Rb0
        PWM1_Init(5000);        //Inizializzazione frequenza segnale pvm
        PWM1_Start();           //Accensione periferica PWM
        OPTION_REG.TOCS=0;      //Disabilito il clock source select bit
        OPTION_REG.PSA=0;       //Prescaler assegnato dal registro timer0 e non dal vatchdog
        OPTION_REG.ps2=0;       // Impostazione prescaler 1:8
        OPTION_REG.ps1=1;       //
        OPTION_REG.ps0=0;       //
        option_reg.intedg=1;    //Settaggio dell'interrupt sul fronte di salita del segnale su RB0
        flag=0;                 //Azzero i due flag
        flag1=0;
```

Fig.10

Figura 10

```
//Inizio Routine modulazione potenza motore avanti indietro
//Porta a2 =1 e porta a3 =0  => Marcia avanti
//Porta a2 =0 e porta a3 =0  => Ferma
//Porta a2 =0 e porta a3 =1  => Marcia indietro e led bianco acceso
if(time >= 224 && time < 300)
{
Porta.F2=1;
Porta.F3=0;
PWM1_Set_Duty(250);           //Potenza massima motore senso orario
flag=0;
}
```

Figura 11

cro tmr0 viene azzerato e di seguito inizia il suo conteggio.
Nel frattempo viene invertito il fronte di rilevamento dell'interrupt, mediante la riga option_reg.intedg=0.
In questo modo al fronte di discesa del segnale di comando, verrà scatenato un nuovo interrupt su rb0, e nello stesso tempo verrà scaricato il valore di tmr0 del pic all'interno della variabile time.
Nella stessa a questo punto, verrà contenuto il valore numerico rappresentativo della durata dell'impulso di comando (vedi paragrafo *descrizione del segnale di comando di un servo per maggior chiarezza*).
A questo punto non rimane che indicare per sommi capi, quali possono essere le modifiche da effettuare, per poter generalizzare il sistema a prescindere dall'ingresso che si ha.
Infatti come già evidenziato, questo sistema è idoneo al comando di un motore in DC in ambo le direzioni di rotazione, ma a fronte di un segnale proveniente da un a scheda di comando di un servo.
Ovviamente non tutti hanno a disposizione un modellino RC, dal quale prelevare tale uscita, quindi mi sembra utile dare una descrizione, di come può essere generalizzato il progetto in modo ad esempio da comandare l'avanzamento il fermo e la rotazione in verso opposto di qualunque motore DC.
Dico qualunque motore DC, in quanto la potenza erogata verso il motore dipenderà solo ed esclusivamente dai MOS utilizzati e dallo stadio di alimentazione esterno, quindi in teoria la nostra scheda di comando sarebbe idonea a pilotare anche motori di elevata potenza.

MODIFICHE AL FIRMWARE

Molto semplicemente possiamo affermare che essendo il segnale PWM responsabile del movimento del motore, per poter pilotare quest'ultimo con due pulsanti, non rimane che escludere l'interrupt su rb0.
Si configurano 3 ingressi ad esempio sulla porta B, che potrebbero essere b1,b2 e b3 dove: b1 fungerà da marcia avanti, b2 da marcia indietro e b3 da arresto di emergenza.

```
//Routine di interrupt
//In questa routine, avviene la misurazione della larghezza dell'impulso sulla porta RB0.
//Nonappena il segnale arriva con il fronte alto del clock, parte l'interrupt ed il contatore tmr0, si attiva.
//Nel frattempo il fronte dell'interrupt viene invertito, così appena arriva il fronte basso del segnale di comando,
//scarico il contenuto del tmr0 nella variabile time.
//In questo modo avrò misura della durata dell'impulso su Rb0
void interrupt() {
if(intcon.intf){
                if(portb.f0==1){
                tmr0=0;
                option_reg.intedg=0;     //Impostazione interrupt, su fronte di discesa

                         }
                if(portb.f0==0){
                time=tmr0;
                tmr0=0;
                option_reg.intedg=1;    //Impostazione interrupt su fronte di salita
                }
                }

                 intcon.intf=0;
                 intcon.gie=1;

}
```

Figura 12

Non rimane che impostare un contatore, che ad ogni pressione dei b1 o b2 incrementa il valore di una variabile la quale dovrà cadere all'interno dei valori presenti negli if (fig.11).
Ovviamente i valori presenti negli if potranno essere cambiati in base alle esigenze, ad esempio: supponiamo di creare una variabile ***contatore*** inizializzata al valore intero 10, la quale ad ogni pressione del pulsante b1 viene incrementata di una unita, mentre alla pressione di b2 viene decrementata.
Alla prima pressione del pulsante la variabile passerà da 10 ad 11, il che determinerà l'avanzamento del motore ad una velocità da noi scelta.
Premendo b2 invece si decrementerà il motore di una quantità di potenza da noi scelta, infatti la variabile contatore passerà da 11 a 10 fermandosi.
Ripremendo il pulsante b2 ad esempio 2 Volte, la variabile assumerà il valore 8 ed il motore inizierà a ruotare nel verso opposto con un'altra velocità scelta.
In questo caso il firmware potrebbe essere modificato come indicato in listato 1.
Bisogna ricordare inoltre di inserire un limite sull'incremento e sul decremento della variabile contatore, ad esempio non farla andare oltre il valore 20 in incremento e al disotto di 0 nel decremento.
Per le modifiche hardware, bisognerà semplicemente aggiungere i 3 pulsanti in corrispondenza delle porte b1,b2 e b3 secondo il seguente schema fig.13.

DESCRIZIONE DEL SEGNALE DI COMANDO DI UN SERVO

Il servo è un dispositivo elettromeccanico, che permette di trasformare una serie di impulsi elettrici di comando, in un movimento meccanico in genere angolare (fig.14 a).

LISTATO 1

Download

```
//All'interno del while//

.
.
.
contatore =10;
.
.
.
While(1) {

if (Button(&PORTB, 1, 1, 1)) {
      oldstate = 1;
}
 if (oldstate && Button(&PORTB, 1, 1, 0)) { // Rilevamento transizione uno_zero
      contatore = contatore++;
      oldstate = 0;                          // Update flag
    }
if (Button(&PORTB, 2, 1, 1)) {
      oldstate = 1;
}
 if (oldstate && Button(&PORTB, 2, 1, 0)) { // Rilevamento transizione uno_zero
      contatore = contatore–;
      oldstate = 0;
if (Button(&PORTB, 3, 1, 1)) {
      oldstate = 1;
}
 if (oldstate && Button(&PORTB, 3, 1, 0)) { // Rilevamento transizione uno_zero
      contatore = 10;                       //Motore fermo
      oldstate = 0;
}

.
.
.
If(contatore==10) {        Porta.F2=0;
                           Porta.F3=0;
                           PWM1_Set_Duty(0);
                           flag=0;

                           }

If(contatore==11) {

                          Porta.F2=1;      //Rotazione oraria
                          Porta.F3=0;
                          PWM1_Set_Duty(50);
                          flag=0;

                          }
If(contatore==12) {

                          Porta.F2=1;      //Rotazione oraria
                          Porta.F3=0;
                          PWM1_Set_Duty(60);
                                        flag=0;

.
.
.
```

```
If(contatore==9) {

                          Porta.F2=0;       //Rotazione antioraria
                          Porta.F3=1;
                          PWM1_Set_Duty(50);
                          flag=0;

                          }
If(contatore==8) {

                          Porta.F2=0;       //Rotazione antioraria
                          Porta.F3=1;
                          PWM1_Set_Duty(60);
                          flag=0;

                          }


Etc etc….
}
```

L'impulso di comando del servo, che in questo caso specifico è il nostro segnale di comando del sistema, è in sostanza un segnale PWM, cha ha un'ampiezza massima pari a 4,5 – 6 Volt, una
frequenza di 50 Hz , ed una larghezza del fronte positivo, che va da 1 ms a 2 ms al massimo per
alcuni, mentre varia da 0,5 ms a 2,5 per altri.
Nel nostro caso, il segnale in uscita al modulo radio, denotava all'oscilloscopio un segnale di 4,8 Volt con una durata minima di 1ms per arrivare ad un massimo di 2,5ms.
Nella fattispecie quindi, quando si dava manetta in avanti con il radiocomando, il segnale allargava la sua durata ed arrivava a 2ms, mentre quando la manetta veniva tirata indietro, il segnale scendeva ad 1ms. Quando la levetta del radiocomando era in posizione di riposo invece, la larghezza del segnale rimaneva fissa ad 1ms.
Nel nostro caso quindi il PIC ha il compito di misurare la durata dell'impulso e capire se: fare andare avanti il motore, accelerare, decelerare, farlo tornare indietro o farlo rimanere fermo.
Il micro 16f876, ha un interessante modulo hardware utilizzabile, che è il CCP ovvero "Capture Compare PWM", il quale permette ad esempio la misurazione dell'intervallo temporale tra due fronti , uno di salita ed uno di discesa.
Questa funzione avrebbe davvero fatto al caso nostro, se non fosse che purtroppo il

Figura 13

Immagine di un Servo generico a) e del segnale di comando b)

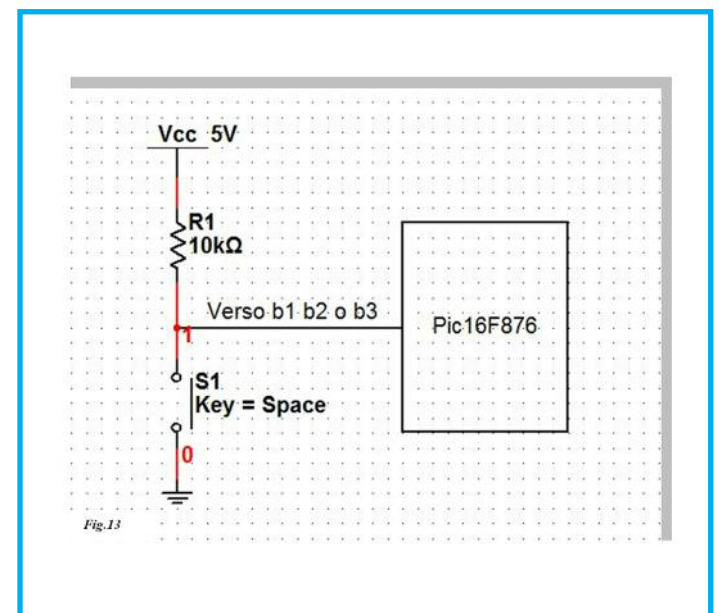
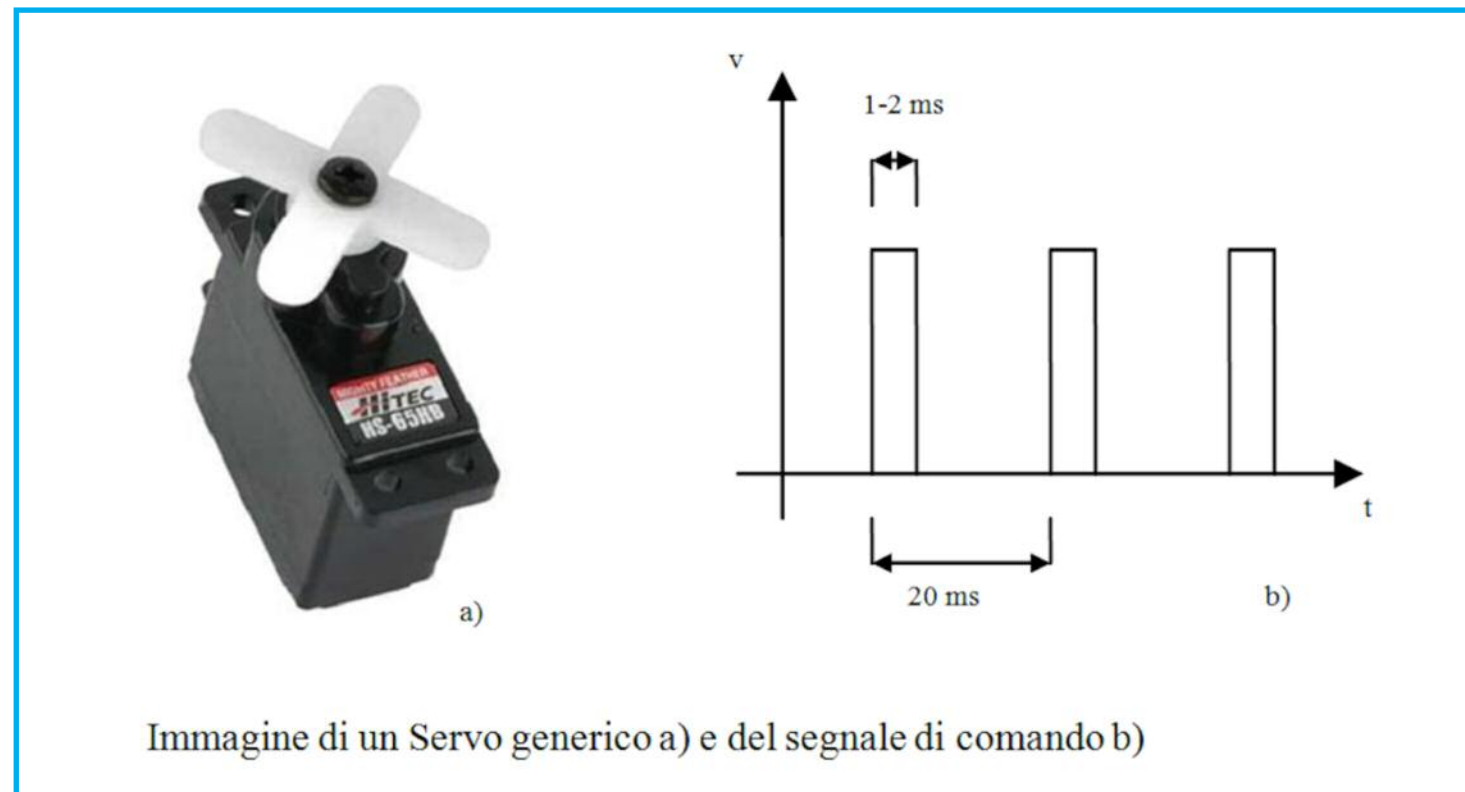



Figura 14

modulo PWM a noi serve per generare il segnale di comando del motore e visto che il modulo PWM del pic in esame può essere usato per l'una o per l'altra funzione, ho deciso di usare quest'ultimo per generare il segnale ed il fronte d'onda di misurarlo mediante un interrupt su rb0.

Elenco Componenti

Commenti	Riferimento	Quantity
Cond.Ceramico	C1, C2	2
Cond.Poliestere	C3	1
Cond.Poliestere	C3A	1
Cond.Elettrolitico	C4, C5, C6, C7, C8, C9	6
Cond.Elettrolitico	C10	1
Cond.Elettrolitico	C11	1
Diodo Fast BA159	D1, D3, D4, D5, D6	5
Diodo Fast BA159	D2	1
LED Verde	DL1	1
Ingresso DC max 20V 6+ 5-	P1	1
Conn Servo	P2	1
LED Bianco	P4	1
LED Arancio	P5	1
IRF540	Q1, Q2, Q3, Q4	4
Resistenza 1/4W	R1	1
Resistenza 1/4W	R2, R3, R4, R5, R6, R7, R8	7
PIC16F876-04/SP	U1	1
SN74LS244N	U2	1
CD4053BCN	U3	1
IR2110	U4, U5	2
7805	U6	1
Quarzo 4 MHZ	Y1	1

di ANGELO DE BARTOLO

POTENZIOMETRI DIGITALI

I potenziometri regolabili digitalmente – noti anche come potenziometri digitali o digPOTs – stanno rimpiazzando le versioni meccaniche "vecchio stile" in tutti i sistemi elettronici. Sono disponibili versioni che hanno differenti risoluzioni, interfacce e prestazioni ma di certo la polvere non sarà più un "rumoroso problema" per regolare il volume

I potenziometri digitali sono molto più affidabili dei potenziometri meccanici, la loro durata si attesta al di sopra dei 50.000 cicli di scrittura contro le poche migliaia di cicli dei corrispondenti meccanici. La risoluzione può variare da pochi bit (di solito 5 bit corrispondenti a 32 step di regolazione) fino a 8 bit ed oltre (256 step di risoluzione) e vengono scelti in base allo specifico impiego, alcuni offrono la possibilità di salvare un dato valore di resistenza come valore di riferimento.
Come per i tradizionali potenziometri anche i potenziometri digitali avranno almeno tre terminali le cui estremità corrispondono al massimo del valore di resistenza ed un terminale intermedio detto "wiper" (spazzola) che in base alla posizione assunta fornisce il valore corrispondente di resistenza, saranno presente i terminali di alimentazione del circuito integrato e i terminali di interfaccia che servono a selezionare un dato valore del wiper. Uno dei parametri fondamentali nella scelta dei potenziometri digitali è l'interfaccia utilizzata: essenzialmente esistono quattro diversi tipologie di potenziometri digitali.

TIPOLOGIA UP/DOWN

I potenziometri di tipo UP/DOWN sono dispositivi il cui valore di resistenza viene selezionato tramite un segnale Up/Down, un segnale di clock ed un segnale di Chip Select. Un esempio commerciale è l'Analog Device AD5220 (schema a blocchi in figura 1) che è provvisto di 8 terminali di collegamento il cui funzionamento è descritto nella tabella 1.

Pin	Nome	Descrizione
1	CLK	Ingresso segnale di clock (trigger sul fronte di discesa)
2	U/D	Ingresso segnale di incremento (U=1) o decremento (U=0).
3	A1	Terminale di resistenza A1
4	GND	Massa di alimentazione
5	W1	Terminale di resistenza W1 a resistenza variabile
6	B1	Terminale di resistenza B1
7	CS	Ingresso Chip Select attivo basso
8	VDD	Positivo di alimentazione (2,7V÷5,5V)

Tabella 1: terminali di collegamento dell'AD5220

FUNCTIONAL BLOCK DIAGRAM

Figura 1: Schema a blocchi dell'AD5220

Il numero di step di risoluzione è facilmente ricavabile dal numero di bit (ovvero gli step) di cui è dotato l'AD5220; essendo la risoluzione a 7 bit corrispondenti a 128 step ed utilizzando la versione da 10k di fondo scala, ogni step vale :

$$R_{STEP} = 10k\Omega / 128 \text{ step} = 78\Omega$$

che sarà appunto il minimo valore di incremento o decremento.

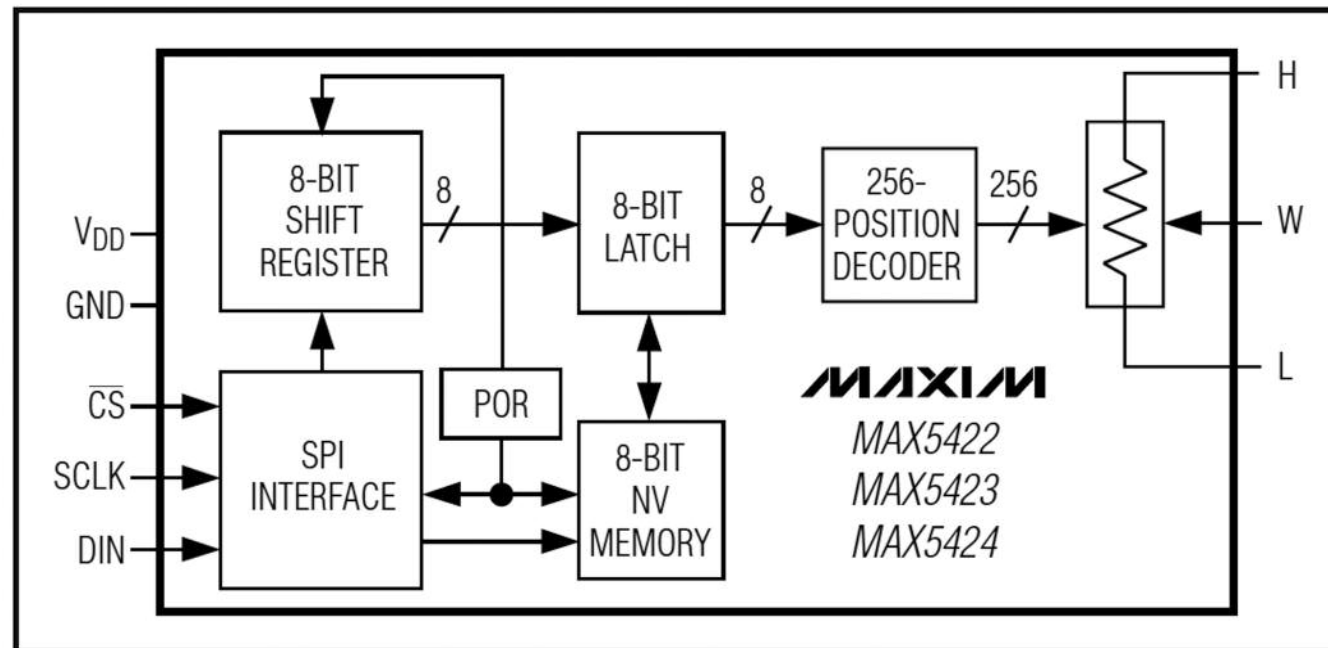



Figura 2: Schema a blocchi del MAX5422

Incremento e decremento negli Up/Down

L'operazione di variazione della posizione del wiper (e quindi della resistenza) si effettua con un impulso di clock (pin 1) durante il quale è necessario mantenere a livello BASSO il pin CS (pin 6), la direzione dell'incremento è controllata dal segnale U/D (pin 2). Se il wiper ha raggiunto l'estremo della resistenza (A o B) eventuali operazione non congrue non avranno effetto (se ad esempio è stato raggiunto il terminale A l'unica operazione possibile è decrementare; se invece è stato raggiunto il terminale B l'unica operazione ammessa è incrementare).

C'è da precisare che al momento dell'accensione il valore prelevabile dal wiper sarà un valore equidistante tra i due terminali (valore centrale) e pertanto l'integrato non mantiene in memoria l'ultimo valore impostato. Bisogna tener presente che l'integrato non prevede alcun dispositivo di debouncing qualora i segnali di ingresso venissero forniti da pulsanti e pertanto bisognerebbe ricorrere a soluzioni esterne (flip-flop, rete RC, oscillatori a colpo singolo "one-shot").
Questa tipologia di potenziometro può essere pilotato da un controllore a logica discreta (come un microcontrollore), da pulsanti meccanici esterni e da encoder rotativi.

0	0	C1	C2	0	0	0	0	D7 MSB	D6	D5	D4	D3	D2	D1	D0 LSB
COMMAND BYTE								DATA BYTE							

Figura 3: Composizione dell'istruzione a 16bit

Register Map

CLOCK EDGE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bit name	—	—	C1	C0	—	—	—	—	D7	D6	D5	D4	D3	D2	D1	D0
Write wiper register	0	0	0	0	0	0	0	0	D7	D6	D5	D4	D3	D2	D1	D0
Write NV register	0	0	0	1	0	0	0	0	D7	D6	D5	D4	D3	D2	D1	D0
Copy wiper register to NV register	0	0	1	0	0	0	0	0	—	—	—	—	—	—	—	—
Copy NV register to wiper register	0	0	1	1	0	0	0	0	—	—	—	—	—	—	—	—

Figura 4: Mappa del registro

TIPOLOGIA SPI

Un potenziometro digitale con interfaccia SPI può essere utilizzato solo con dispositivi dotati di interfaccia SPI hardware o software, ad esempio i microcontrollori. Un esempio commerciale è il MAXIM MAX5422 (schema a blocchi in figura 2) a 8 bit di risoluzione e provvisto di 8 terminali (descrizione in tabella 2) con interfaccia SPI a 3 fili con memoria non volatile del posizione di wiper registrata che verrà richiamata all'avvio del dispositivo.

Settaggio della posizione del wiper

Per settare un particolare valore nel potenziometro è necessario inviare una sequenza di 16 bit (word) composta da un *Command Byte* e da un *Data Byte* (figura 3).
Come si evince dalla mappa del registro di figura 4 ci sono 4 possibili operazioni effettuabili selezionando opportunamente i bit C1 e C0 e sono:

- impostazione della posizione del wiper;
- salvataggio della posizione del wiper nella memora non volatile;
- copiare il valore della posizione del wiper nella memoria non volatile;
- impostare il valore della posizione del wiper con il valore presente nella memoria non volatile.

Un'operazione di WRITE sarà necessariamente a 16 bit (8 bit per il command-byte e 8 bit per il data-byte) mentre un'operazione di copia potrà essere a 8 bit (solo command-byte) oppure a 16 bit in cui gli 8 bit dei dati non verranno considerati dal dispositivo. Un tipico diagramma di timing è raffigurato in figura 5.
Il costruttore dichiara garantita la memoria non volatile (contenente il valore della po-

Pin	Nome	Descrizione
1	VDD	Positivo di alimentazione (2,7V÷5,25V)
2	SCLK	Interfaccia SPI: input segnale di clock
3	DIN	Interfaccia SPI: input segnale dati
4	CS	Interfaccia SPI: chip select attivo basso
5	GND	Massa di alimentazione
6	L	Terminale di resistenza L
7	W	Terminale di resistenza W a resistenza variabile
8	H	Terminale di resistenza H

Tabella 2: terminali di collegamento del MAX5422

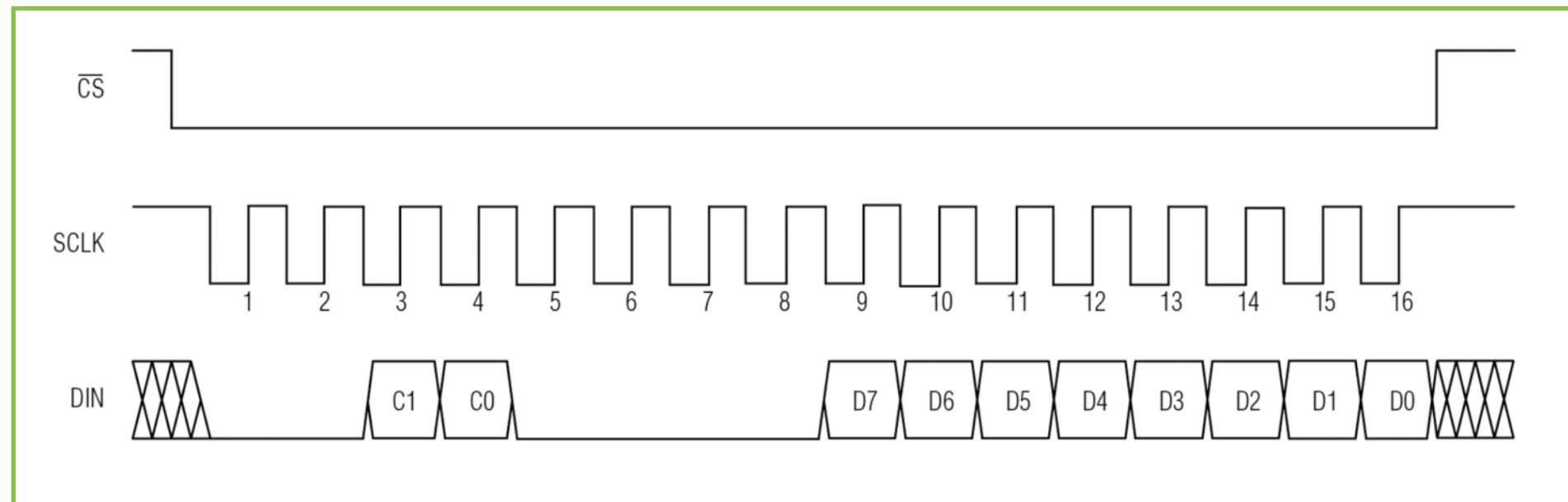



Figura 5: Diagramma di timing

sizione di wiper salvata) per 50 anni e fino a 200.000 cicli di scrittura.
Alcuni dispositivi di potenziometri digitali con interfaccia SPI possono avere delle modalità di utilizzo diverse da quelle descritte: ad esempio il CAT5172 della ON Semiconductor necessita solo di un byte in ingresso composto dagli 8 bit che identificano il livello di resistenza voluto in una scala da 0 a 255 (0 corrisponde ad una resistenza quasi nulla – tipicamente di 50 ohm, 255 corrisponde al valore massimo di resistenza – fondo scala). Altri dispositivi possono avere due o più potenziometri integrati come ad esempio il MCP42010 della MicroChip in cui è necessario specificare tramite Command-Byte quale potenziometro si vuole usare (a tale scopo vengono utilizzati gli ultimi 2 bit del Command-Byte).
L'uso di questa tipologia di potenziometri è da ricercarsi esclusivamente con controller dotati di interfacce SPI hardware o software.

TIPOLOGIA I2C

I potenziometri con interfaccia I2C sono costituiti essenzialmente da un blocco destinato alla gestione del protocollo I2C, un blocco di alimentazione ed il blocco relativo alla resistenza variabile (schema a blocchi in figura 6). Il circuito integrato commercializzato da Maxim MAX5417 è un dispositivo con risoluzione a 8 bit, provvisto di 8 terminali (descrizione in tabella 3) con un pin esterno (pin 4 – A0) utilizzabile per settare lo Slave Address del dispositivo.

Pin	Nome	Descrizione
1	VDD	Positivo di alimentazione (2,7V÷5,25V)
2	SCL	Interfaccia I²C: input segnale di clock
3	SDA	Interfaccia I²C: input segnale dati
4	A0	Interfaccia I²C: bit di indirizzo del dispositivo
5	GND	Massa di alimentazione
6	L	Terminale di resistenza L
7	W	Terminale di resistenza W a resistenza variabile
8	H	Terminale di resistenza H

Tabella 3: terminali di collegamento del MAX5417

L'indirizzo del dispositivo (detto appunto Slave Address) è standard a 7 bit in cui i primi 4 bit (A6…A3) sono fissi e valgono "0101", i successivi 2 bit (A2, A1) sono programmati in fabbrica in base ad una lettera posta come suffisso (vedi figura 7) e l'ultimo (A0) è impostabile tramite un semplice ponticello verso massa o positivo d'alimentazione.

Comunicazione con il Master

Per poter comunicare con il MAX5417 è necessario che il Master invii dei byte che possano essere correttamente interpretati dal dispositivo. La comunicazione è tipica del protocollo I2C ed inizia con un segnale di START da parte del Master, un primo byte contenente lo *Slave Address* a 7 bit e l'ottavo bit posto a 0, un secondo byte contenente l'operazione - *Command Byte* - ed un eventuale ultimo byte contenente i

Figura 6: Schema a blocchi del MAX5417

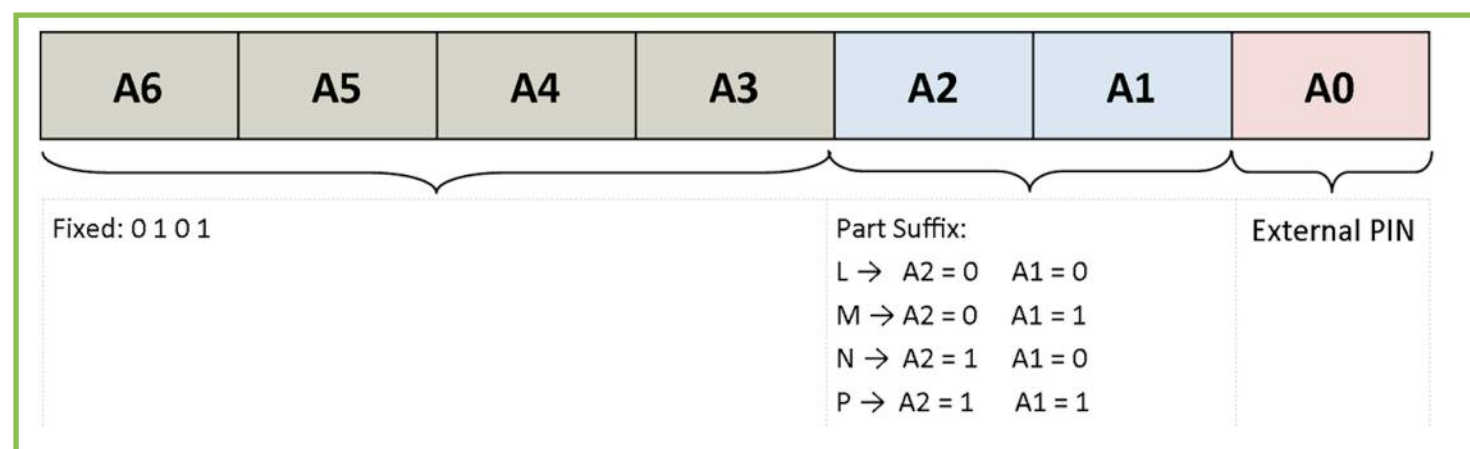



Figura 7: Composizione dello slave address a 7 bit del MAX5417

Comando	Control byte								Descrizione
VREG	0	0	0	1	0	0	0	1	Il "data byte" viene scritto nella memoria volatile ed il wiper assume il valore della memoria volatile.
NVREG	0	0	1	0	0	0	0	1	Il "data byte" viene scritto nella memoria non volatile (il wiper non cambia la posizione).
NVREGxVREG	0	1	1	0	0	0	0	1	Copia il valore della memoria non volatile nella memoria volatile e viene aggiornata la posizione del wiper.
VREGxNVREG	0	1	0	1	0	0	0	1	Copia il valore della memoria volatile nella memoria non volatile (il wiper non cambia).

Tabella 4: Command Byte del MAX5417

dati - *Data Byte*: un esempio di scrittura di un nuovo valore del wiper è visibile in figura 8. I possibili comandi sono riepilogati nella tabella 4.
All'avvio il dispositivo assumerà il valore che è contenuto nella memoria non volatile. Questa tipologia di potenziometri è utilizzabile esclusivamente con controller dotati di interfacce I²C hardware oppure software.

START	0	1	0	1	A2	A1	A0	0	ACK	0	0	0	1	0	0	0	1	ACK	D7	D6	D5	D4	D3	D2	D1	D0	ACK	STOP

SLAVE ADDRESS — FROM MAX5417 — COMMAND BYTE — FROM MAX5417 — DATA BYTE — FROM MAX5417

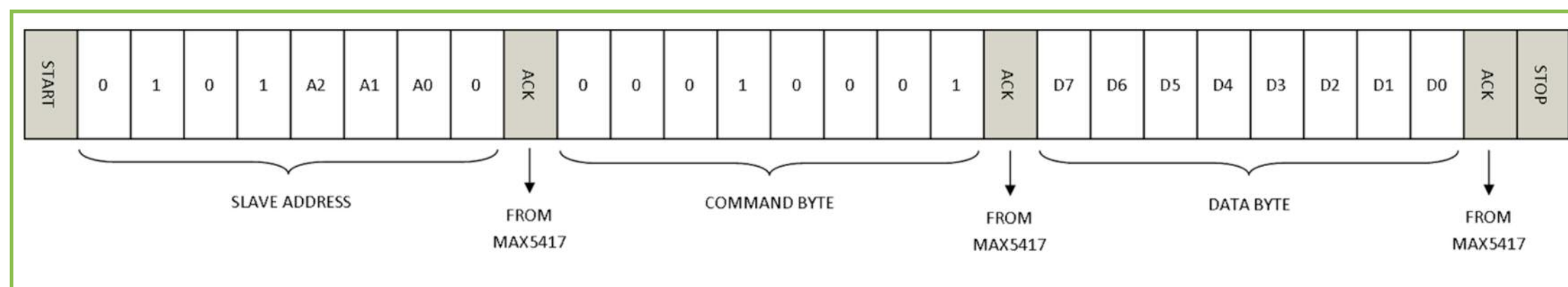



Figura 8: Struttura di un comando di scrittura del wiper register

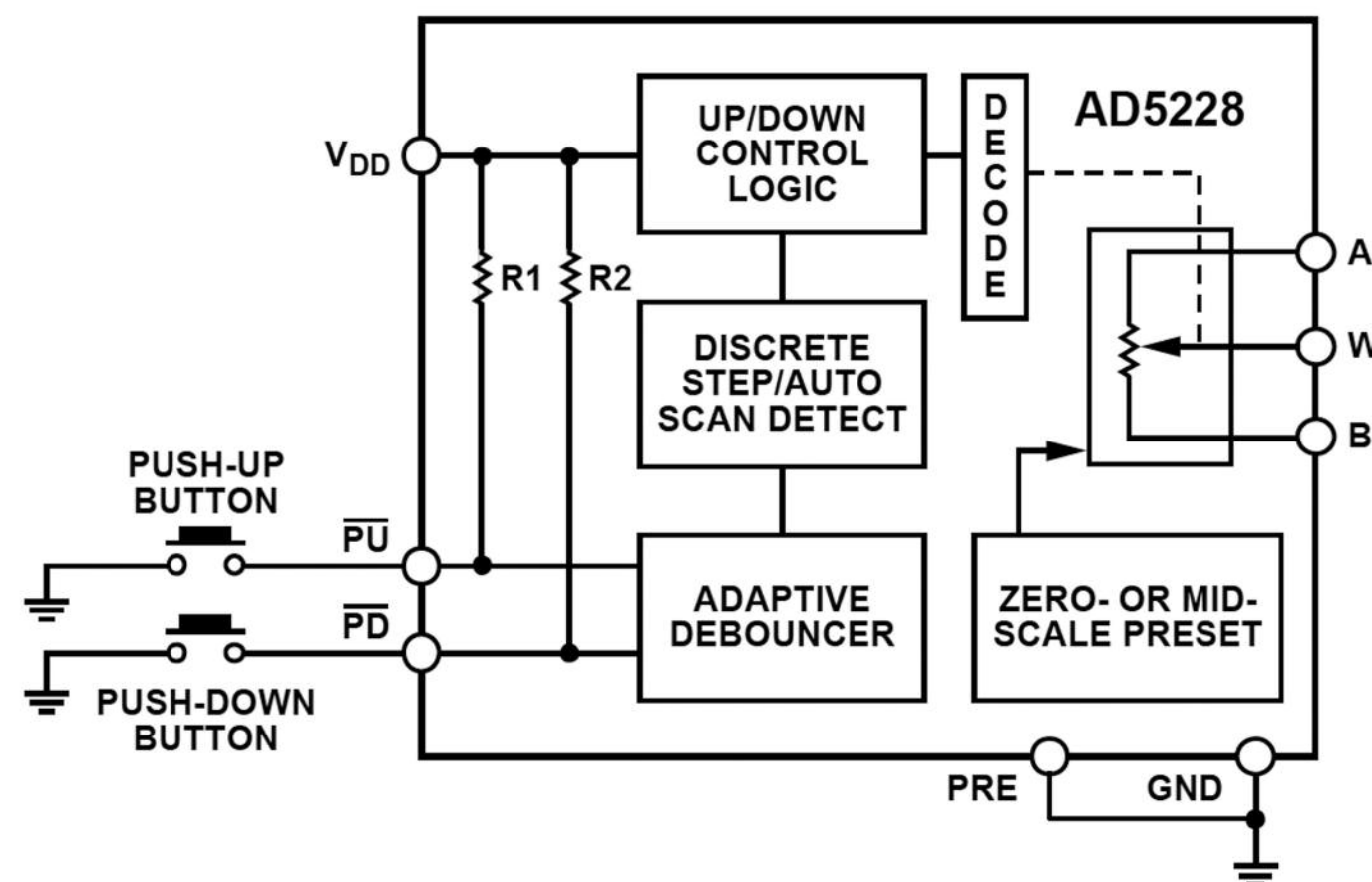



Figura 9: Schema a blocchi dell'AD5228

TIPOLOGIA PUSH BUTTON

Questa tipologia di potenziometro digitale è la più semplice da utilizzare e ricorda maggiormente i tradizionali potenziometri rotativi. Un esempio commerciale è l'Analog Device AD5228 a 5 bit di risoluzione i cui 8 terminali sono descritti nella tabella 5. Lo schema a blocchi (figura 9) suggerisce che l'utilizzo di questo componente è molto semplice e non richiede componenti aggiuntivi oltre all'alimentazione ed ai due pulsanti di incremento e decremento.

Funzionamento e debouncing

E' da notare che questo integrato contiene all'interno un sistema adattativo di antirimbalzo applicato sui terminali corrispondenti ai pulsanti (pin 1 e pin 2). Questo sistema verifica se un livello basso ad uno dei due pin è mantenuto per almeno 11ms: se è

Pin	Nome	Descrizione
1	PU	Pin del pulsante di incremento con resistenza interna di pull-up da 100kΩ
2	PD	Pin del pulsante di decremento con resistenza interna di pull-up da 100kΩ
3	A	Terminale di resistenza A
4	GND	Massa di alimentazione
5	W	Terminale di resistenza W a resistenza variabile
6	B	Terminale di resistenza B
7	PRE	Power-On-Preset. Impostato a GND il valore del wiper all'avvio del dispositivo sarà un valore centrale. Impostato a VDD il wiper avrà un valore di inizio scala.
8	VDD	Positivo di alimentazione (2,7V÷5, 5V)

Tabella 5: terminali di collegamento dell'AD5228

più breve allora è interpretato come un rimbalzo e quindi non effettua alcuna operazione mentre se è maggiore viene completata l'operazione corrispondente al pulsante premuto. A titolo d'esempio l'Analog Device dichiara che un giocatore di videogames avanzato riesce e premere e rilasciare un pulsante in un tempo non inferiore a 40ms; ovviamente se si pilotano i pin tramite microcontrollore e non tramite pulsanti esterni bisogna prestare attenzione a mantenere il livello basso per almeno 12ms. Mantenendo premuto un pulsante per più di un secondo si entra nella modalità di *Autoscan* la quale comporta un incremento o decremento di 4 step al secondo (uno step ogni 0,25 secondi).

Di sicuro è la tipologia più semplice da usare in quanto non necessita né di interfaccia dati né di elementi circuitali aggiuntivi (spesso non sono necessarie neanche le resistenze di pull-up in quanto presenti internamente) e possono essere controllati sia da pulsanti meccanici che da microcontrollori.

COME SCEGLIERE?

Uno dei parametri più critici nella scelta di un potenziometro digitale è proprio l'interfaccia di comunicazione. Le caratteristiche da tener conto sono pertanto:

- *Interfaccia di comunicazione (Up/Down, SPI, I²C, PushButton);*
- *Controller del wiper (microcontrollore, pulsanti);*
- *Capacità di memorizzazione;*
- *Valore fondo scala;*
- *Risoluzione;*
- *Prestazioni (tolleranza, coefficiente di temperatura, banda passante, distorsione armonica);*
- *Package.*

CONCLUSIONI

I potenziometri digitali hanno un gran vantaggio riscontrabile nell'affidabilità, spazio ridotto, rumori elettronici di gran lunga minori rispetto ai corrispondenti meccanici. I digPOTs possono sostituire i potenziometri meccanici praticamente in qualsiasi applicazione consentendo al progettista ed all'utente di trarne beneficio.

a cura della redazione

LED-VOLTMETRO

Un circuito dotato di una barra di LED che si accendono in sequenza in base al valore della tensione continua applicata all'ingresso. Un ottimo esercizio pratico sui comparatori di tensione

Lo schema elettrico per la realizzazione del dispositivo richiesto è riportato nella figura 1 ed impiega un circuito integrato LM3914 che da solo provvede alla gestione dei LED facendo in modo che essi si accendano in sequenza a seconda del valore della tensione continua applicata all'ingresso.
Prima di addentrarci all'interno del LM3914, analizziamo la funzione dei pochi componenti esterni. La tensione da misurare viene applicata al trimmer P1 il cui valore è stato scelto piuttosto elevato per non assorbire troppa corrente dalla sorgente di segnale e garantire quindi una elevata impedenza di ingresso. Dal cursore del trimmer si preleva una porzione del segnale di ingresso tanto maggiore quanto più il cursore è spostato verso il punto A. Da questo di deduce che P1 regola la sensibilità dello strumento. Il diodo D1 protegge il sistema da eventuali inversioni di polarità. C1 funziona come elemento di "zavorra" per la tensione, in questo modo

Figura 1: Schema del voltmetro a LED

Figura 2: Forme d'onda relative allo stadio di ingresso

ai capi di C1 avremo una tensione continua e stabile. R4 consente al condensatore di scaricarsi in modo da seguire l'andamento della eventuali variazioni della tensione di ingresso. D2 è un diodo zener da 5,1V che è stato inserito per proteggere il circuito dalle sovratensioni di ingresso. La presenza di D2 fa sì che la tensione che arriva all'ingresso del LM3914 venga tagliata se supera i 5,1V. R3 ha il solo scopo di limitare la corrente in D2 nel caso si attivi il meccanismo di protezione visto in precedenza.
Per chiarire ulteriormente il funzionamento dello stadio di ingresso appena descritto, si veda la figura 2 in cui sono riportate le forma d'onda nel caso in cui il cursore di P1 si trovi nel punto A. In verde la tensione di ingresso, mentre in rosso è riportata la tensione V2 in ingresso al LM3914. L'esempio riportato in figura prevede l'applicazione di una tensione non continua all'ingresso, per evidenziare che lo strumento misura in ogni caso il valore massimo positivo di quanto applicato.
Si noti l'effetto di D2 che taglia tutte le tensioni superiori a 5,1V. Le resistenze R1 ed R2 permettono di ottenere una tensione di riferimento necessaria al LM3914 sfruttando una generatore di tensione interno al circuito integrato stesso. Tale tensione è disponibile al piedino 7 (REF OUT) ed il suo valore è dato da 1,25(1+R2/R1).
Con i valori scelti per R1 ed R2 si ottiene una tensione di riferimento pari a 5V. La tensione così ottenuta viene applicata al pin 6 (RHI) e costituisce la soglia di accensione del LED10. La resistenza R1 ha anche lo scopo di limitare la corrente nei LED che vale I_{LED}=12,5/R1.
All'interno del LM3914 vi sono dieci comparatori come mostrato in figura 3. A ciascun comparatore è applicata la tensione di ingresso tramite il buffer (amplificatore non invertente con amplificazione unitaria) e una tensione di riferimento pari ad un certo numero di decimi della tensione applicata al pin 6 RHI.
In particolare, il comparatore che pilota il LED1 ha, come tensione di riferimento, un decimo della tensione RHI (0,5V nel caso del circuito di figura 1), quello che pilota il LED2 ha 2 decimi (1V) e così via. Il pin 9 (MODE) consente di selezionare il modo di funzionamento del VU-Meter.
È possibile operare in tre modalità diverse: Bar Graph Display (visualizzazione come barra luminosa), Dot Display Single (visualizzazione come punto luminoso), Dot Display (visualizzazione come punto luminoso su 20 o più LED – in questa modalità sono necessari più di un modulo LM3914).
Per il funzionamento in Bar Graph si deve collegare il pin 9 al positivo di alimentazione (caso di figura 1), per il Dot Display Single il pin 9 deve essere lasciato disconnesso, mentre per il Dot Display multiplo il pin 9 del primo modulo al pin 1 del secondo e così via.

Figura 3: I blocchi all'interno dell'integrato LM3914

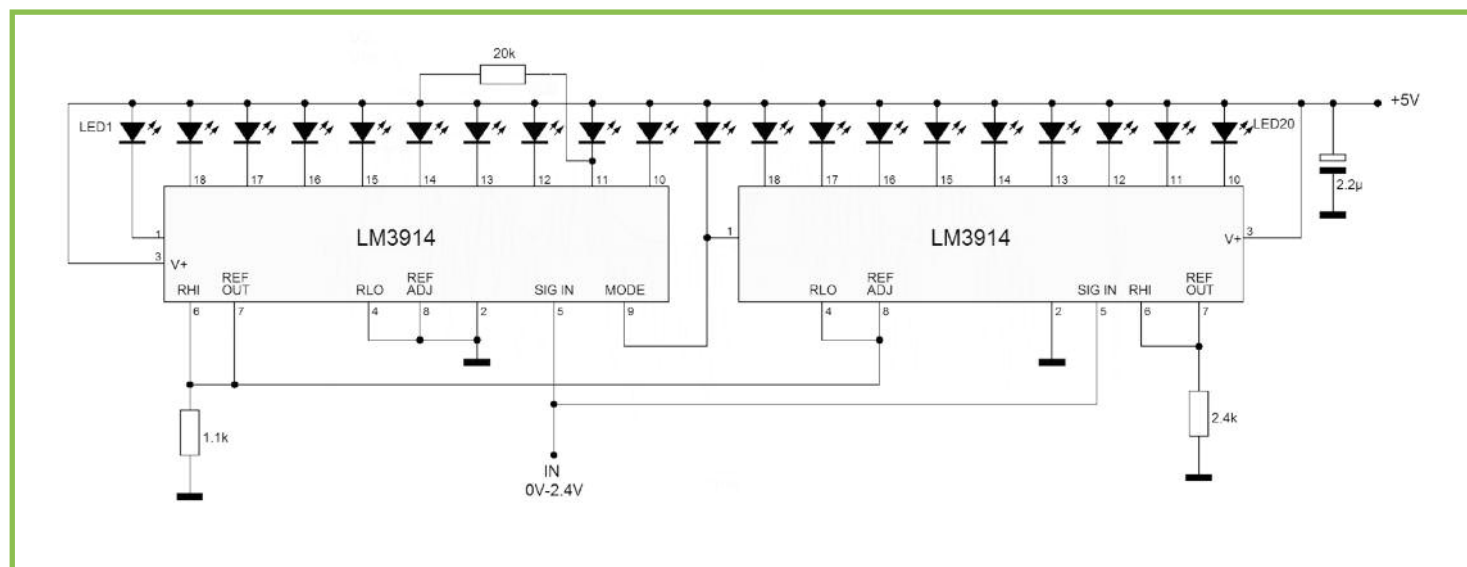



Figura 4: Voltmetro con 20 LED e visualizzazione a punto luminoso

In figura 4 è riportato una realizzazione del voltmetro su 20 LED con visualizzazione a punto luminoso. Il circuito può essere impiegato anche come Vu-meter, collegando l'ingresso direttamente all'uscita di un amplificatore audio.

COMPARATORE

Un comparatore è in grado di riconoscere se una tensione applicata al suo ingresso è maggiore o minore di una soglia di riferimento. Il modo più semplice per realizzare un comparatore è l'impiego di un amplificatore operazionale senza la resistenza di reazione. Idealmente l'amplificazione differenziale è infinita quindi in uscita avremo una tensione positiva (pari circa al valore della tensione di alimentazione positiva) se la tensione al morsetto invertente è minore di quella al morsetto non invertente. Viceversa se la tensione al morsetto invertente è maggiore di quella al morsetto non invertente, la tensione di uscita sarà circa uguale a quella di alimentazione negativa. Fissando quindi il valore di una delle due tensioni ad un valore di riferimento V_R è possibile, dall'analisi della tensione di uscita, sapere se sull'altro morsetto vi è una tensione maggiore o minore di quella di riferimento.
Un comparatore è quindi realizzabile secondo lo schema di figura 5. In figura 6 sono riportate le forme d'onda relative al circuito di figura 5 nel caso in cui la tensione di ingresso sia una sinusoide di ampiezza 2V e la tensione di riferimento sia V_R=1V (quando non diversamente specificato, l'operazionale si intende alimentato con una tensione simmetrica ±Vcc di 15V). Si noti che quando Vin è maggiore di VR la tensione di uscita è bassa (-15V), mentre quando Vin è minore di V_R, la tensione di uscita è alta (+15V).
Ovviamente è possibile applicare la tensione di riferimento al morsetto invertente (figura 7) ottenendo, a parità di segnale di ingresso, le forme d'onda di figura 8. Si noti che all'interno del LM3914 vi sono dieci comparatori come quello di figura 5 in cui la tensione di riferimento è ottenuta dal pin 6 (RHI) mediante un partitore di 10 resistenze uguali.

Figura 5: Comparatore realizzato con un amplificatore operazionale

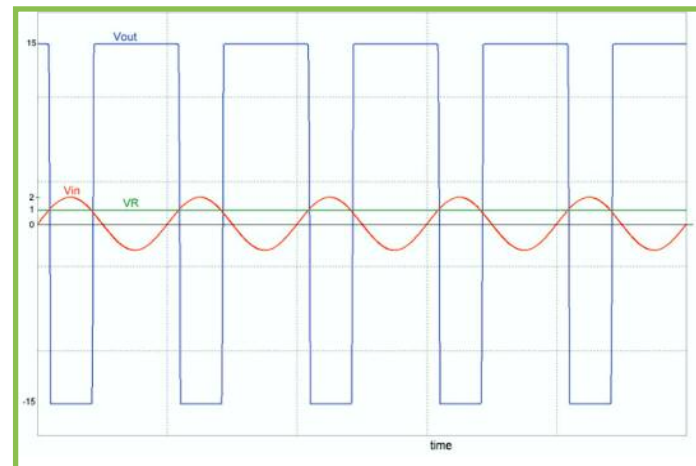



Figura 6: Le forme d'onda per il comparatore di figura 5

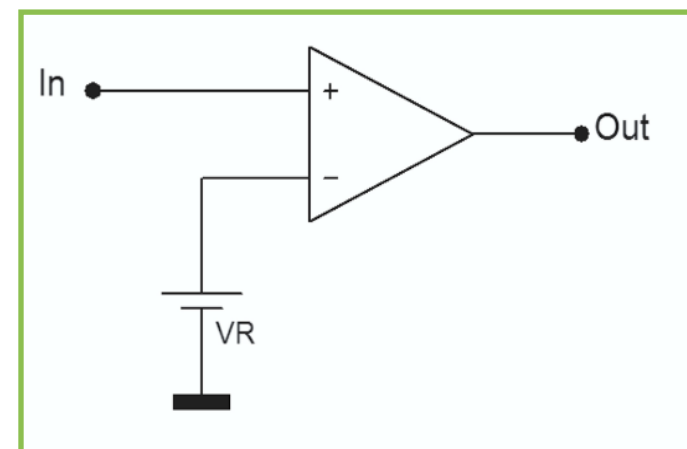



Figura 7: La tensione di riferimento è applicata al morsetto invertente

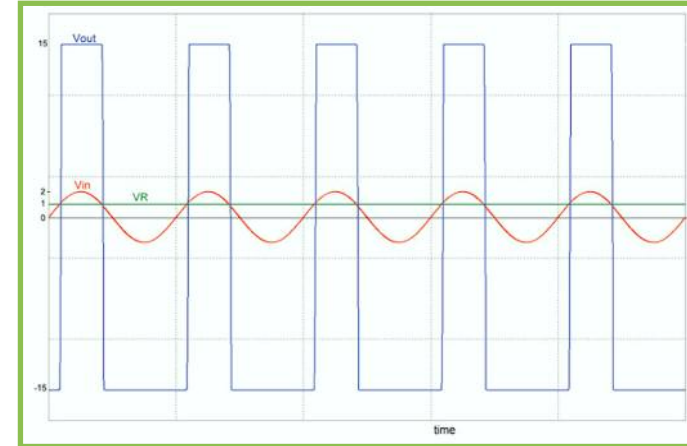



Figura 8: Le forme d'onda per il comparatore di figura 6

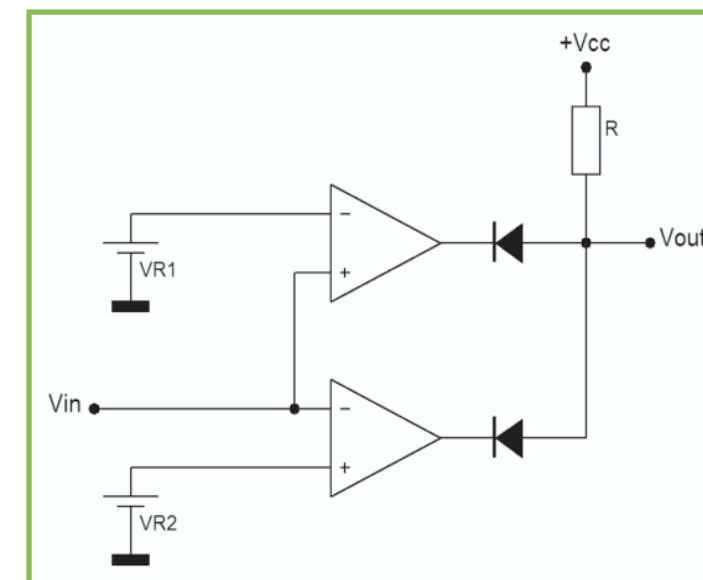



Figura 9: Comparatore a finestra: l'uscita è alta solo se l'ingresso è nel range dei valori di riferimento

Comparatore a finestra

Può presentarsi la necessità si sapere quando la tensione di ingresso rientra all'interno di due valori prefissati (finestra). A tale scopo è possibile utilizzare il cosiddetto "comparatore a finestra" riportato in figura 9. In questo caso la tensione di uscita è alta solo se la tensione di ingresso è compresa tra V_R1 e V_R2 (ovviamente $V_R1<V_R2$) come mostrato in figura 10.
Modificando il circuito come indicato in figura 11 si ottiene un comparatore a finestra la cui uscita è alta se l'ingresso è minore di V_R2 o maggiore di V_R1 ($V_R2<V_R1$). Una applicazione del circuito di figura 11 potrebbe essere ad esempio nel monitoraggio del livello di un liquido in un serbatoio: in questo modo è possibile azionare una segnalazione (acustica o luminosa) se il livello del liquido è al di fuori di un intervallo prestabilito (MINMAX).

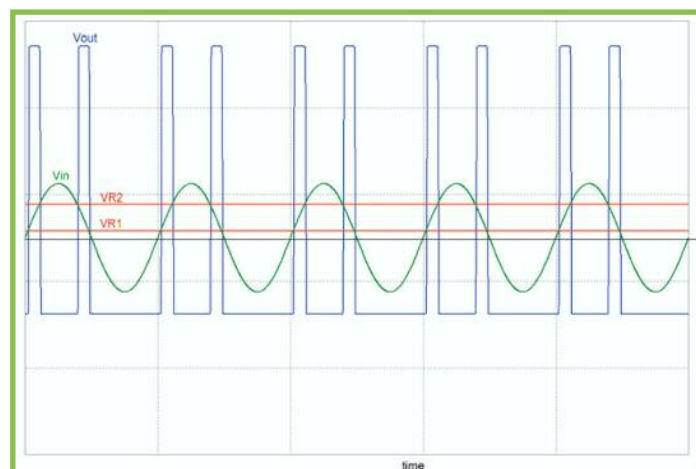



Figura 10: Le forme d'onda per il comparatore a finestra di figura 9

Comparatore con isteresi

I comparatori sono molto utilizzati in campo digitale per "ripulire" i segnali ed avere quindi fronti di salita e discesa ben definiti. Un segnale digitale proveniente da un sensore è infatti affetto da disturbi e raramente è utilizzabile direttamente. Se i livelli logici "alto" e "basso" corrispondono rispettivamente a 5V e 0V, potrei allora tentare di ripulire il segnale utilizzando un semplice comparatore e fissando la tensione di soglia a 2,5V: in questo modo tutto ciò che è al di sotto di 2,5V verrà riconosciuto come livello logico basso, mentre tensioni superiori a 2,5V saranno riconosciute come livello logico alto.

In figura 12 è riportato un segnale in uscita da un sensore (Vs) e lo stesso segnale all'uscita del comparatore avendo fissato 2,5V come tensione di riferimento.

Figura 12: Un segnale digitale "ripulito" con un comparatore

Si noti che nonostante il segnale sia stato ripulito, la zona tratteggiata in grigio risulta piuttosto critica in quanto il segnale di ingresso compie delle piccole oscillazioni attorno al valore di soglia. È possibile ovviare al problema impiegando un comparatore con isteresi. Questo è un particolare tipo di comparatore cha ha due tensioni di soglia: una valida quando il segnale di ingresso è in salita, l'altra quando il segnale di ingresso è in discesa. Il comparatore con isteresi è detto anche trigger di Schmitt e la sua realizzazione è riportata in figura 13.

Figura 13: Comparatore con isteresi: Trigger di Schmitt

Come si può notare, la tensione di soglia non è fissa, ma è ottenuta mediante la partizione della tensione di uscita, quindi se l'uscita è +Vcc, la tensione di soglia vale V_H=VccR1/(R1+R2) mentre se l'uscita è –Vcc, la tensione di soglia vale V_L=-VccR1/(R1+R2). Per comprendere meglio il funzionamento supponiamo che l'ingres-

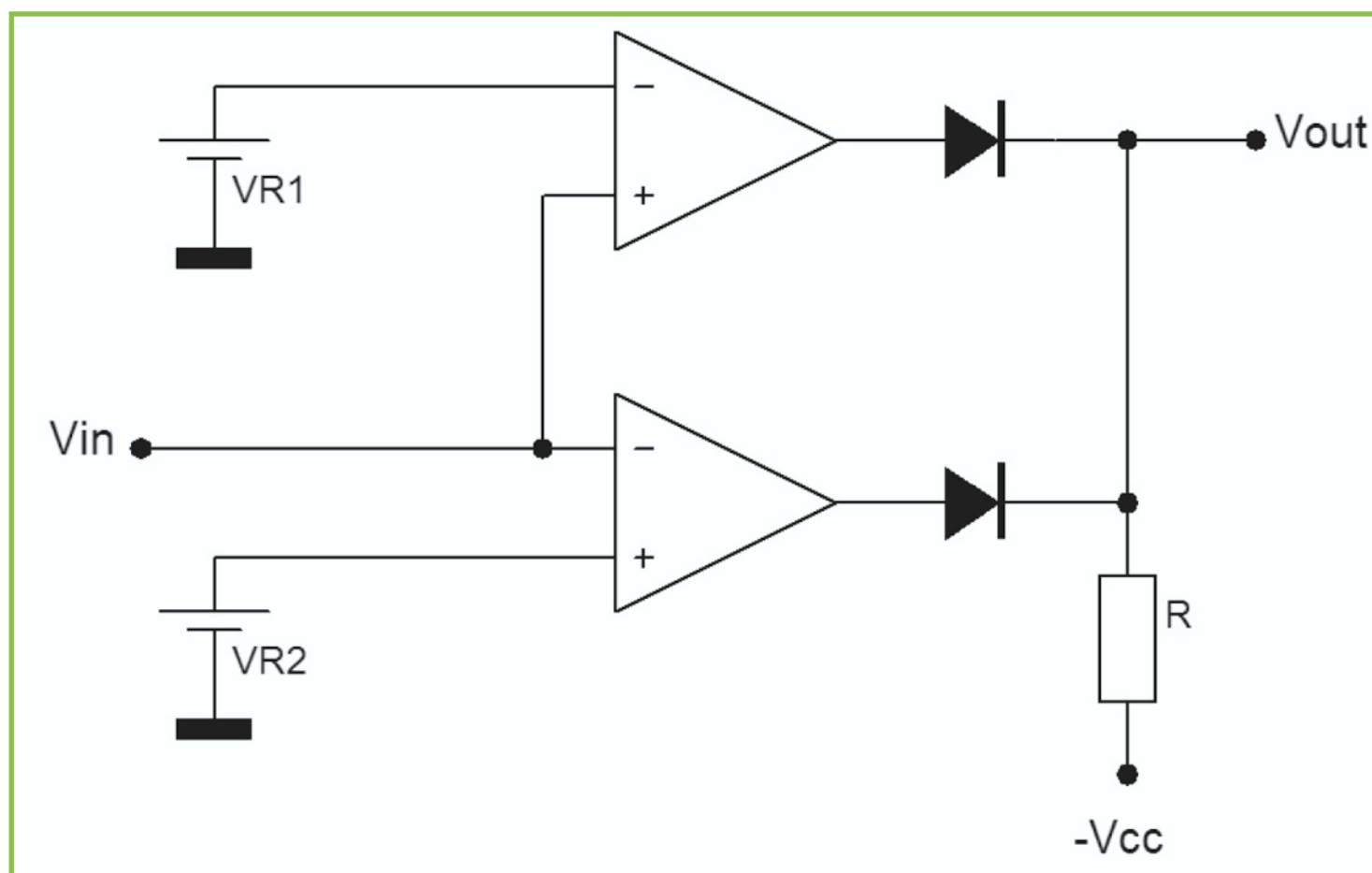



Figura 11: Altro tipo di comparatore a finestra: l'uscita è alta se l'ingresso è fuori dal range dei valori di riferimento

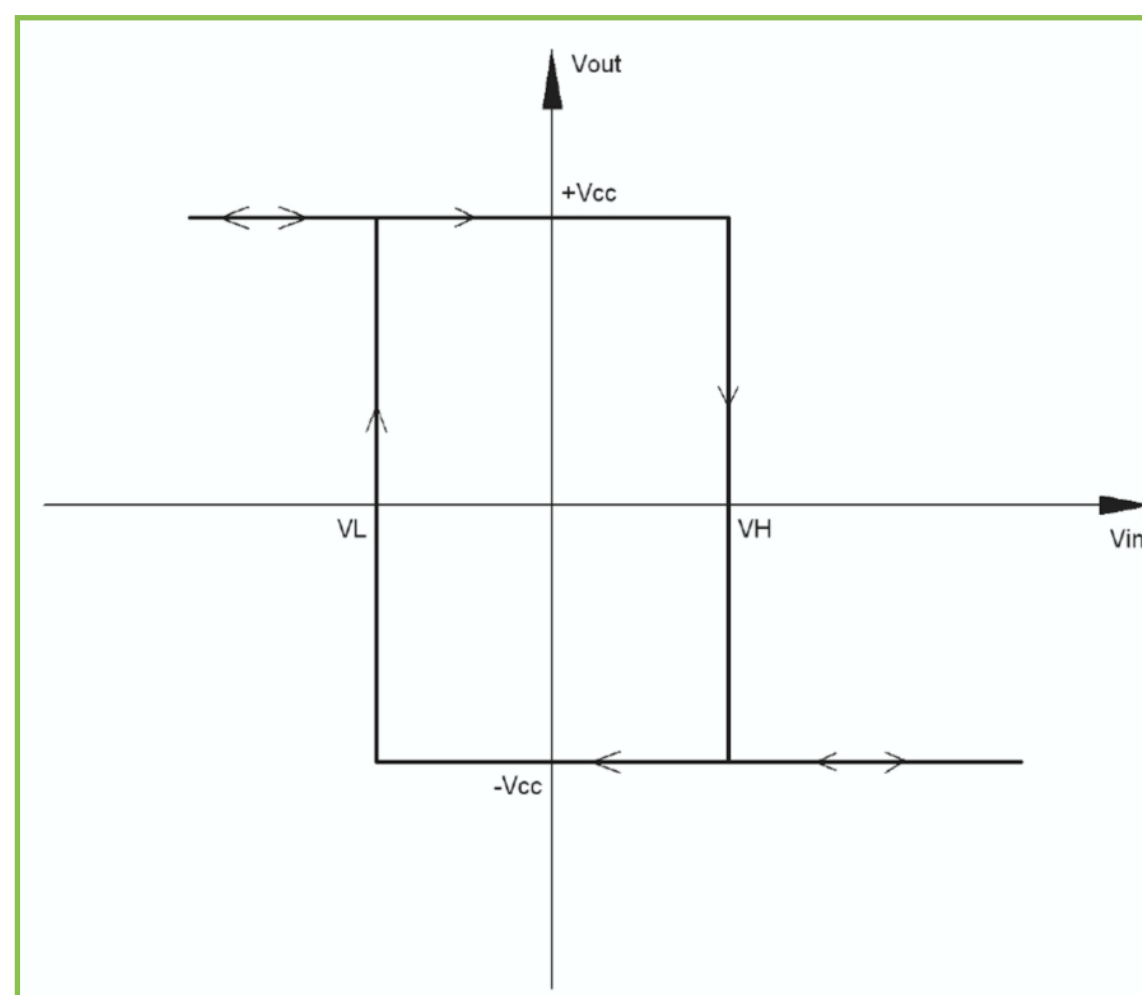



Figura 14: Caratteristica ingresso-uscita del trigger di Schmitt

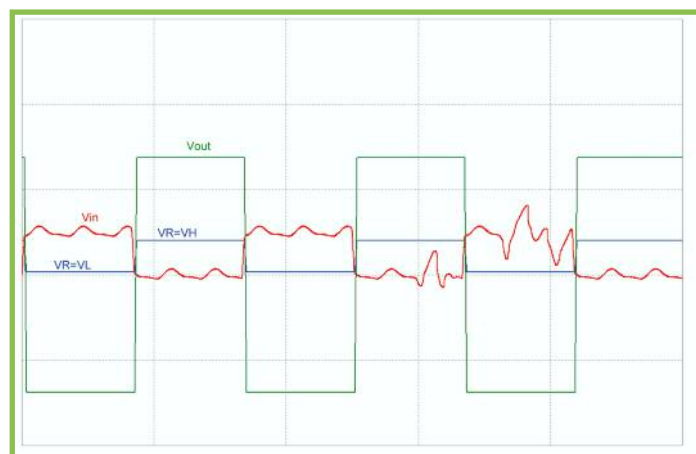



Figura 15: Un segnale digitale "ripulito" con un trigger di Schmitt

so sia ad un valore molto basso, inferiore alla tensione di soglia: l'uscita sarà alta (+Vcc) e la tensione di soglia sarà V_H. Aumentando il segnale di ingresso il sistema resterà in questo stato fintanto che Vin non supera V_H.
A questo punto l'uscita diviene bassa (-Vcc) e la tensione di soglia vale ora V_L. Diminuendo il segnale di ingresso, lo stato del sistema resta invariato finché Vin non scende al di sotto di V_L. Riportando in un grafico l'andamento della tensione di uscita al variare dell'ingresso si ottiene il classico diagramma ad isteresi riportato in figura 14. Vediamo allora come questo circuito risulti migliore del precedente per il trattamento dei segnali digitali spuri. Fissando le tensioni di soglia ai valori V_H=4,5V e V_L=0,5V la situazione di figura 12 diviene quella di figura 15.
Si noti che in questo caso viene eliminata definitivamente la zona di incertezza tratteggiata in figura 12 benché il segnale di ingresso sia affetto da disturbi anche più intensi. Il trigger di Schmitt riportato in figura 13 è invertente ed infatti in figura 15 si vede che se Vin è alto, Vout è basso e viceversa.
Per quanto detto in merito alle tensioni V_H e V_L è evidente che con il circuito di figura 13 non è possibile ottenere due tensioni di riferimento non simmetriche. Tuttavia è possibile traslare la caratteristica di figura 14 verso destra o verso sinistra applicando un generatore di tensione al partitore resistivo come illustrato nella figura 16. Se V1 è positivo, la caratteristica si sposta verso destra della quantità V1, altrimenti si sposta verso sinistra della stessa quantità.

Figura 16: Metodo per rendere asimmetriche le tensioni di soglia

METTETEVI ALLA PROVA

Alla luce di quanto detto a proposito dei comparatori, vi proponiamo un esercizio di progettazione.
Avendo a disposizione una fotoresistenza lineare il cui valore è 1Kohm in assenza di luce e 100 ohm in piena luce, sapreste realizzare un interruttore crepuscolare dotato di regolazione per la sensibilità che sia in grado di eccitare un relé se la luminosità scende al di sotto del livello di sensibilità impostato?

IEshop

RISONATORE INDUSTRIALE A 50 MHZ IN FORMATO DIP

Questo risonatore a 50 MHz (RoHS Compliant) è un risonatore consigliato quando si implementa un progetto OEM DIP compatibile con il Basic Stamp 2sx , ma può anche essere utilizzato per i microcontrollori della serie SX. Il risonatore ceramico con capacitanza built-in, permette una facile applicazione su vari progetti, senza la necessità di un adeguamento per l'oscillazione.

 Prezzo: € 0.99

ARDUINO ROBOT

Arduino Robot è il primo Arduino ufficiale su due ruote. Il robot è dotato di due processori, uno su ciascuna delle sue due schede PCB. La scheda motore controlla i motori, e la scheda di controllo legge i sensori e decide come operare. Ciascuno dei pannelli è una scheda Arduino completa programmabile utilizzando l'IDE di Arduino. Sia la scheda motore che quella di controllo sono schede a microcontrollore basati sulla ATMEGA32U4.

BESTSELLER

Il robot ha molti dei suoi perni mappati ai sensori di bordo e attuatori.

La programmazione di Arduino Robot è simile a quella di Arduino Leonardo. Entrambi i processori sono dotati di comunicazione USB, eliminando la necessità di un processore secondario. Questo permette al robot di apparire a un computer collegato a una porta seriale / COM virtuale (CDC). Come sempre con Arduino, ogni elemento della piattaforma - hardware, software e documentazione - è liberamente disponibile e open-source.

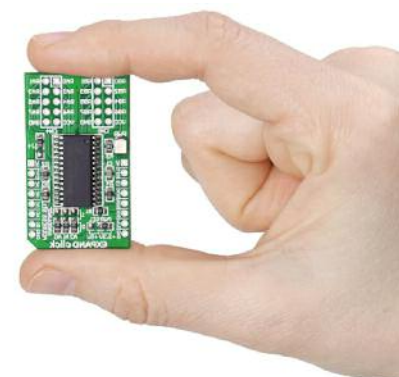
Questo significa che si può imparare esattamente come è fatto e utilizzare il design come punto di partenza per i propri robot. Arduino Robot è il risultato dello sforzo collettivo di un team internazionale guardando a come la scienza può essere divertente per imparare.

 Prezzo: € 240.79

EXPAND CLICK

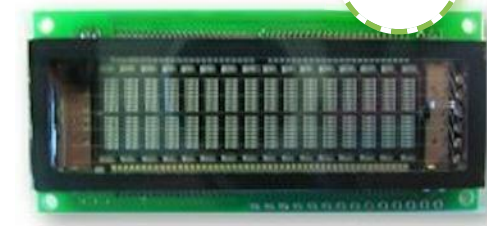
L'EXPAND Click è una scheda accessoria che utilizza il nuovo socket mikroBUS. Include un expander I/O MCP23S17 a 16 bit con velocità di clock SPI fino a 10 MHz, per applicazioni di throughput più elevato. Tre jumper permettono di configurare l'indirizzo della scheda e collegare fino a otto dispositivi sul bus. Il controller supporta anche funzionalità di interrupt-on-change. La scheda è impostata per utilizzare l'alimentazione 3.3V di default., ma è presente un jumper SMD saldato PWR SEL sulla posizione 5V da utilizzare con i sistemi a 5V.

NOVITÀ

 Prezzo: € 10.89

DISPLAY 16X2 LCD EMULATOR VFD FUTABA

OFFERTA

Questo display emula il comportamento di un classico display LCD 16x2 per cui può essere utilizzato in sostituzione di un classico LCD senza fare alcuna modifica al progetto.

 Prezzo: € ~~47.19~~ € 42.46

ARDUINO YÚN

Arduino YÚN è il primo di una nuova linea innovativa di prodotti wifi che combinano la potenza di Linux con la facilità d'uso di Arduino.

Il primo Arduino Yun è la combinazione di un classico Arduino Leonardo (basato sul processore ATMEGA32U4) con un sistema-one-chip WiFi esecuzione Linino (un MIPS GNU / Linux basata su OpenWRT). Abbiamo incorporato la macchina Linux direttamente sul PCB della Arduino Leonardo collegando i due in modo tale da facilitare l'esecuzione di comandi su Linux tramite Arduino e poterlo usare come interfaccia Ethernet e Wifi.

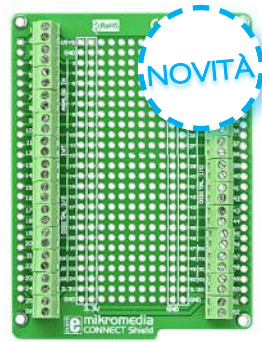
 Prezzo: € 64.13

RACCOLTA ANNATE 2006/2012 DI FIRMWARE

Un file zip che contiene tutte le riviste dal 2006 al 2012 in formato PDF ad alta risoluzione, tutti i downloads realtivi ai singoli numeri. E' incluso anche Acrobat Reader 6 per sfogliare le riviste.

 Prezzo: € 20.16

MIKROMEDIA CONNECT SHIELD

La Connect shield è una scheda di espansione pin-compatibile con tutte le schede Mikromedia che permette di stabilire facilmente le connessioni tra la scheda Mikromedia e l'elettronica circostante. Sulla scheda sono presenti i morsetti a vite ad alta qualità che consentono di collegare ogni pin in modo rapido e senza saldature. Questo è particolarmente utile in fase di sviluppo di prototipi, ma permette anche di integrare la schede Mikromedia come prodotto finale. L'area breadboard di prototipazione è disponibile anche per l'inserimento di componenti aggiuntivi ed espandere le funzionalità di base con caratteristiche personalizzate. La scheda viene fornita con due connettori femmina 1x26 pin.

 Prezzo: € 14.52

RS485 CLICK 5V

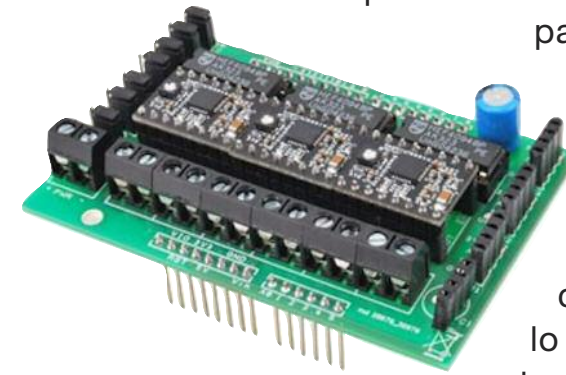
RS485 Click 5V è una scheda accessoria che utilizza il nuovo socket mikroBUS. Il ricetrasmettitore di linea differenziale on-board ADM485 è adatto per comunicazione dati bidirezionale ad alta velocità sulle linee bus di trasmissione multipunto. La scheda è stata progettata per la trasmissione dati equilibrata ed è conforme agli standard EIA RS-485 e RS-422. Questo ricetrasmettitore dispone di un driver e un ricevitore di linea differenziale. La scheda opera solo con alimentazione a 5V.

 Prezzo: € 16.34

BEE CLICK

BEE Click è una scheda accessoria che utilizza il nuovo socket mikroBUS. È dotato di un modulo radio transceiver MRF24J40MA a 2,4 GHz IEEE 802.15.4. Questo modulo include un'antenna integrata sul PCB ed è collegato al microcontrollore attraverso un'interfaccia SPI. Come tale, questo modulo è una soluzione ideale per le reti wireless, smart home automation, building automation e altre applicazioni elettroniche che richiedono la comunicazione wireless. La scheda è stata progettata per usare solo l'alimentazione 3.3V.

 Prezzo: € 25.41

SHIELD ARDUINO PER MOTORI

Shield per Arduino Uno o Mega, che permette di comandare tre motori passo-passo bipolari da 2A. Il controllo dei motori è gestito da un driver (MD09B - uno per motore) prodotto dalla Pololu. Ogni driver contiene un doppio ponte ad H a MOSFET e può essere impostato per gestire sia la direzione, sia il numero di gradi che il rotore del motore deve compiere alla ricezione di ogni comando; in altre parole, possiamo decidere se quando lo comandiamo, il modulo deve far ruotare l'albero di uno step alla volta, oppure di 1/2, 1/4, 1/8 o 1/16, in base all'accuratezza che si desidera ottenere. Sulla shield sono presenti anche 4 ingressi analogici.

 Prezzo: € 53.00

ARDUINO ON ANDROID KIT

NOVITÀ

Usa la tua Arduino MEGA ADK (con cavo usb incluso) per connettere un dispositivo Android. Monta tutti i tuoi esperimenti sulla breadbourd e unisci i tuoi componenti usando il tuo wire kit. Costruisci degli indicatori visuali alcuni dei 40 LEDs inclusi nel kit; ce ne sono in 4 colori differenti. Costruisci il tuo controller fisico per cellulare usando alcuni potenziometri (ne ottieni 10 in totale). Fai un semplice robot usando 2 servo motori a rotazione continua. Misura l'intensità di luce usando alcuni dei 5 sensori luminosi inclusi nel kit. Rappresenta testi o piccole grafiche sulle 2 display a matrice di led bicolore (32x16 LEDs). Controlla il tuo modno fisico con il modulo relay (comprensivo di cavi). Verifica che il tuo progetto non è sottosopra usando il tilt module. Fai una piccola tastiera con i 5 pulsanti. Fai suonare piccole melodie usando ilpiezo speaker. Misura le distanze usando il sensore a ultrasuoni. Alimenta i tuoi progetti con un regolatore di voltaggio a 5V. Determina l'ammontare di gradi centigradi usando ilsensore di temperatura . E non dimenticare che hai anche un kit con 176 resistors per i tuoi prototipi. Questo kit è stato creato da D. Cuartielles and A. Goransson per coloro che intendono seguire gli esempi presentati nella Phase 1 del libro Professional Android Open Accessory Programming with Arduino per la Wrox professional series. Con questo kit è possibile riprodurre tutti gli esempi tra cui LED display per ui tuoi SMS, sensori di parcheggio, un semplice robot, un campionatore di suoni.

 Prezzo: € 229.78

DISPLAY LCD TFT 4,3” PER BEAGLE BONE BLACK - VERSIONE TOUCH RESISTIVO

Il 4DCAPE-43T è specificamente progettata per la Beagle Bone Black (BBB), e fornisce un display da 4,3” per la BBB per l'interazione diretta dell'utente e la visualizzazione delle informazioni. Disponibile sia in touch resistivo (4DCAPE-43T) e non-touch (4DCAPE-43), il CAPE dispone di un display TFT LCD da 4,3" 480x272 di risoluzione e sette pulsanti. La Beagle Bone Black si collega direttamente alla parte posteriore del 4DCAPE-43, e fornisce tutti i segnali richiesti, come il segnale del display e l’alimentazione. La scheda comprende anche quattro fori di fissaggio per un facile montaggio. Si prega di notare che la 4DCAPE-43 è stato progettata per essere il display primario solo per Beagle Beagle Bone Black e non può essere essere programmata utilizzando sistemi 4D Systems Workshop 4 IDE.

Prezzo: € 59.29

- Sei già cliente? Risparmierai il 10% sul tuo nuovo ordine!

- Il tuo ordine supera i 200 EUR? Le spese di trasporto sono Gratis!

- Vuoi ricevere un buono sconto di 5 EUR? Recensisci i prodotti acquistati!

- Vuoi ricevere particolari offerte o promozioni? Diventa membro Inware Edizioni